

LEVEL

2

NAVAL POSTGRADUATE SCHOOL

Monterey, California

AD A102662



DTIC
ELECTED
AUG 11 1981

DTIC FILE COPY

THESIS

ENGINEERING DESIGN OPTIMIZATION
BY THE AUGMENTED LAGRANGE MULTIPLIER METHOD

by

10 Leroy E. Madsen

March 1981

12/90

Thesis Advisor:

G. N. Vanderplaats

Approved for public release; distribution unlimited

81 8

251/50
11 046

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Engineering Design Optimization by the Augmented Lagrange Multiplier Method		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; March 1981
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Leroy E. Madsen		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE March 1981
		13. NUMBER OF PAGES 89 pages
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Augmented Lagrange Multiplier Method, engineering design optimiza- tion, equality and inequality constraints. structural design optimization, linear convergence.		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A computer program was developed for solving equality and in- equality constrained optimization problems by the Augmented Lagrange Multiplier method. The program was developed specifically for use in engineering design. The historical evolution and theoretical development of the multiplier method is presented. Several examples are used to demonstrate the effects of penalty parameters and multipliers on		

DD FORM 1473

EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE/When Data Entered

Item 20 (contd)

the convergence and accuracy of the method. Computational experience with variations to the method is documented.

A brief literature search of the multiplier method's application to engineering design is summarized. The method is demonstrated with several mathematical and engineering examples. A comparison to classical penalty methods and the method of feasible directions was performed in each case.

Accession For	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A	

Approved for public release; distribution unlimited

Engineering Design Optimization
by the Augmented Lagrange Multiplier Method

by

Leroy E. Madsen
Lieutenant, United States Navy
B.S.E.E., Purdue University, 1974

Submitted in partial fulfillment of the
requirements for the degree of
MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
March 1981

Author:

Leroy E. Madsen

Approved by:

Samuel N. Vanderplank
Thesis Advisor

David Salinas
Second Reader

R. J. Marts
Chairman, Department of Mechanical Engineering

William M.oller
Dean of Science and Engineering

ABSTRACT

A computer program was developed for solving equality and inequality constrained optimization problems by the Augmented Lagrange Multiplier method. The program was developed specifically for use in engineering design.

The historical evolution and theoretical development of the multiplier method is presented. Several examples are used to demonstrate the effects of penalty parameters and multipliers on the convergence and accuracy of the method. Computational experience with variations to the method is documented.

A brief literature search of the multiplier method's application to engineering design is summarized. The method is demonstrated with several mathematical and engineering examples. A comparison to classical penalty methods and the method of feasible directions was performed in each case.

TABLE OF CONTENTS

I.	INTRODUCTION-----	6
II.	THE MULTIPLIER METHOD-----	9
	A. BACKGROUND-----	9
	B. THE LAGRANGIAN-----	10
	C. PENALTY FUNCTION METHODS-----	12
	D. THE EQUALITY CONSTRAINED PROBLEM-----	17
	E. THE INEQUALITY CONSTRAINED PROBLEM-----	29
III.	COMPUTATIONAL ASPECTS-----	36
	A. THE UNCONSTRAINED MINIMIZATION PROBLEM-----	36
	B. THE ONE-DIMENSIONAL SEARCH-----	41
	C. THE MULTIPLIER METHOD-----	44
IV.	COMPUTATIONAL EXPERIENCE-----	48
	A. CASE 1: THE CONSTRAINED ROSEN-SUZUKI FUNCTION---	48
	B. CASE 2: THE SIMPLE QUADRATIC FUNCTION-----	49
	C. CASE 3: THE PAVIANI FUNCTION-----	50
	D. CASE 4: THE THREE-BAR TRUSS PROBLEM-----	50
	E. CASE 5: THE CANTILEVER BEAM PROBLEM-----	52
	F. SUMMARY-----	52
V.	CONCLUSION-----	60
	LIST OF REFERENCES-----	62
	APPENDIX A: One-Dimensional Search Algorithm-----	64
	APPENDIX B: Sample Computer Output-----	66
	APPENDIX C: Computer Listing-----	72
	INITIAL DISTRIBUTION LIST-----	89

I. INTRODUCTION

Several methods currently exist which solve the function minimization problem. While each has its advantage in specialized cases, no one method, including the multiplier method, is universally more efficient and accurate than the others. The Augmented Lagrange Multiplier method has been shown to be an efficient and accurate method of engineering design optimization. The multiplier method is particularly advantageous in nonlinear equality constrained problems. It has become increasingly popular in engineering design optimization because of its good convergence rate and its attractive theoretical properties.

The general nonlinear problem is defined:

$$\text{Minimize } f(\bar{X}) \quad (1.1a)$$

$$\text{subject to } g_i(\bar{X}) \leq 0 \quad (i=1, \dots, \ell) \quad (1.1b)$$

$$\text{and } h_j(\bar{X}) = 0 \quad (j=1, \dots, m < n) \quad (1.1c)$$

$$\text{and } X_i^l < X_i < X_i^u \quad (i=1, \dots, n) \quad (1.1d)$$

where \bar{X} is a vector of n design variables, $f(\bar{X})$ is the objective function, $g(\bar{X})$ is a set of ℓ inequality constraint functions, and $h(\bar{X})$ is a set of m equality constraint functions, and X_i^l and X_i^u are bounds on the design variables, referred to as side constraints. The problem is solved by creating a single augmented Lagrange function $L(\bar{X}, \bar{\lambda})$, where $\bar{\lambda}$ is a vector of n Lagrange multipliers. The optimal solution $(\bar{X}^*, \bar{\lambda}^*)$ is

found by alternately solving a series of unconstrained minimizations followed by a simple update of the multipliers, $\bar{\lambda}$.

In this study, the unconstrained minimization is performed by the Davidon-Fletcher-Powell method [1] utilizing a combination of the Golden Section and polynomial one-dimensional search procedures. The Davidon-Fletcher-Powell method provides good reliability and convergence properties. The Golden Section search provides a reliable means of reducing the search bounds. In most cases, polynomial interpolation obtains a more accurate minimum than Golden Section for sufficiently narrow bounds. The combination provided a reliable and accurate one-dimensional search.

The multiplier method has significant advantages. As will be shown later, the inherent numerical ill-conditioning of more common penalty function methods is reduced in the multiplier method. Exact solutions, not possible in penalty methods, are also attainable. Finally, multiplier methods are not restricted to convex programming as are pure primal-dual methods.

There have been numerous applications of the multiplier method in the various disciplines of engineering design optimization. Imai [2] effectively used the method in structural optimization. Fax and Mills [3], while limited to equality constraints, applied the method to heat exchanger optimization. Hedderich's [4] work in heat exchanger

optimization showed the need for a method which could treat equality constraints directly, in addition to inequality constraints. He showed the Constrained function Minimization Program, CONMIN [5] to be very inefficient for the equality constrained problem. It is his work that motivated this research of the multiplier method.

It is the objective of this research to develop an operational computer program applicable to the various disciplines of engineering design optimization. It is not the author's intent to develop a universally superior optimization program, but to develop a program to be incorporated into a library of optimization programs with flexibility in the choice of unconstrained minimization subprograms, one dimensional search subprograms, derivative evaluation subprograms, and convergence criterion.

The theory and historical development of the multiplier method is presented, concluded by a multiplier method algorithm for nonlinear equality and inequality constrained problems. Experimentation with computational aspects is then summarized. Finally, a set of mathematical and engineering test cases are solved demonstrating the method's effectiveness, followed by conclusions and a discussion of results.

II. THE MULTIPLIER METHOD

The development of the multiplier method and a brief mathematical background is presented in this chapter. Mathematical proofs are brief since computational aspects and application of the method are more the concern of this thesis. This chapter includes historical background of the method, a review of the Lagrange multiplier and penalty function methods, and a development of the algorithm for equality and inequality constrained problems. Rigorous mathematical development of the method is given by Hestenes [6].

A. BACKGROUND

The multiplier method was independently introduced in 1968 by Hestenes [7] and Powell [8]. Both developed an augmented Lagrange function and solved a series of unconstrained minimizations followed by a simple update of the multiplier vector. Powell showed that, if the function had continuous second derivatives, the method would converge locally at a linear rate while the penalty parameter remained finite. As will be shown later, this eliminated the inherent numerical ill-conditioning of classical penalty function methods.

Extensive testing and modification has been done to the method of Hestenes and Powell. Miele et. al. [9,10] applied the method extensively to equality constrained problems. Tripathi and Navandra [11] experimented with updating the multiplier after each one dimensional search instead of after each unconstrained minimization. Rockafellar [12,13] expanded the method to nonlinear inequality constrained problems. Global convergence of the method was proved by Rockafellar [13] for convex programming problems. Bertsekas [14,15] provides convergence proofs for the general nonlinear problem.

B. THE LAGRANGIAN

The general nonlinear equality constrained problem is defined:

$$\text{Minimize } f(\bar{X}) \quad (2.1a)$$

$$\text{subject to } h_i(\bar{X}) = 0 \quad (i=1, \dots, m < n) \quad (2.1b)$$

The theorem associated with Lagrange multiplier method states that:

"If \bar{X}^* affords a local minimum to $f(\bar{X})$ subject to the constraints $h_i(\bar{X})=0$, then there exists a unique set of multipliers, λ_i , ($i=1, \dots, m$) such that if

$$L(\bar{X}, \bar{\lambda}) = f(\bar{X}) + \sum_{i=1}^m \lambda_i h_i(\bar{X}), \quad (2.2a)$$

then

$$\nabla L(\bar{X}^*, \bar{\lambda}^*) = \nabla f(\bar{X}^*) + \sum_{i=1}^m \lambda_i \nabla h_i(\bar{X}^*) = 0 \quad (2.2b)$$

and

$$L''(\bar{X}^*, \bar{\lambda}^*) = \frac{\partial^2 L(\bar{X}^*, \bar{\lambda}^*)}{\partial^2 \bar{X}_i} > 0 \quad (2.2c)$$

where ∇ denotes the gradient of the function and L''

denotes the second derivative of L'' [6].

Equations 2.2b and 2.2c are the necessary conditions for locally constrained minima. Equation 2.2b and the feasibility condition (Eq. 2.1b) constitute the Kuhn-Tucker necessary conditions for optimality [6]. It is assumed that $f(\bar{X})$ and $h_i(\bar{X})$ are second order differentiable and that the gradients $\nabla h_i(\bar{X})$ are not zero at \bar{X}^* .

The problem can now be stated in terms of the equivalent classical Lagrangian.

$$\text{Minimize } L(\bar{X}, \bar{\lambda}) \quad (2.3a)$$

$$\text{subject to } h_i(\bar{X}) = 0 \quad (i=1, \dots, m < n) \quad (2.3b)$$

Assuming the existence of the saddle points of the Lagrangian $L(\bar{X}, \bar{\lambda})$, the condition exists:

$$L(\bar{X}^*, \bar{\lambda}) \leq L(\bar{X}^*, \bar{\lambda}^*) \leq L(\bar{X}, \bar{\lambda}^*) \quad (2.4)$$

The optimal pair $(\bar{X}^*, \bar{\lambda}^*)$ can be obtained by first minimizing $L(\bar{X}, \bar{\lambda})$ respect to \bar{X} , then maximizing $L(\bar{X}^k, \bar{\lambda})$ with respect to λ by update equation,

$$\lambda_i^{k+1} = \lambda_i^k + c[h_i(\bar{X}^k)] \quad (2.5)$$

where c is a scalar parameter (stepsize), k is the iteration number, and \bar{X}^k is the local minimum of $L(\bar{X}^k, \bar{\lambda}^k)$. The procedure is repeated until convergence is attained. This is the so called "primal-dual" method.

Serious disadvantages are encountered in the primal-dual method. First, the problem (Eq. 2.3) must have a locally convex structure for the dual problem to be well defined and

for Equation 2.5 to be meaningful [16]. Second, a large number of iterations are usually required to minimize $L(\bar{X}, \bar{\lambda})$ (Eq. 2.2a) since the ascent iteration (Eq. 2.5) converges only moderately fast. Thus, primal-dual methods have found application in only a limited class of problems where minimization of the Lagrangian (Eq. 2.2a) can be efficiently carried out due to special structure, as shown by Luenberger [16], or where the design problem exhibits a unique form, as shown by Schmit and Fleury [17].

C. PENALTY FUNCTION METHODS

Penalty function methods have been used extensively since the mid-1940's [18]. They are considered to be efficient for inequality constrained problems.

Given the nonlinear inequality constrained problem:

$$\text{Minimize } f(\bar{X}) \quad (2.6a)$$

$$\text{subject to } g_i(\bar{X}) \leq 0 \quad (i=1, \dots, \ell) \quad (2.6b)$$

The general exterior penalty function is defined:

$$F_e(\bar{X}, c) = f(\bar{X}) + c \sum_{i=1}^{\ell} \phi(t) \quad (2.7)$$

where $\phi(t)$ is some scalar penalty function of the constraints and c is some scalar penalty parameter. The most common penalty function is the quadratic $\phi(t)=t^2/2$. However, it may be desirable at times to use other penalty functions. In this study the quadratic penalty function is used such that Equation 2.7 becomes

$$F_e(\bar{X}, c) = f(\bar{X}) + \frac{c}{2} \sum_{i=1}^l p_i^2(\bar{X}) \quad (2.8)$$

$$\text{where } p_i(\bar{X}) = \begin{cases} g_i(\bar{X}), & \text{if } g_i(\bar{X}) \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

The problem is now an unconstrained minimization of $F_e(\bar{X})$. The convergence of the method is easily seen by a simple numerical example.

Example 1 [2]. Minimize x such that $1-x \leq 0$.

Solution: First, examine the solution to the problem by primal-dual method. Equation 2.2a becomes

$$L(x, \lambda) = x + \lambda(1-x) \quad (2.9)$$

where λ is a real non-negative number. From the stationary conditions of Equation 2.9:

$$(x^*, \lambda^*) = (1, 1)$$

Solving by exterior penalty method, Equation 2.8 becomes

$$F_e(x, c) = x + \frac{c}{2} p^2(x) \quad (2.10)$$

$$\text{where } p(x) = \begin{cases} 1 - x, & x \geq 1 \\ 0, & \text{otherwise} \end{cases}$$

From the stationary conditions, a minimum exists at

$$x' = 1 - \frac{1}{c}$$

To obtain the optimal x^* , a series of unconstrained minimizations are solved while increasing the penalty parameter c toward infinity. It is apparent that as $c \rightarrow \infty$, $x' \rightarrow x^*$. No exact solution is obtained and the optimum is approached from the infeasible region.

The plots of $F_e(x)$ vs. x and $f(x)$ vs. c are given in Figs. 1 and 2, respectively. Note in Fig. 1, as c gets larger, the function changes more rapidly and the optimum becomes more difficult to find regardless of the minimization technique. This is a cause of numerical ill-conditioning inherent with penalty methods. Figure 2 shows the asymptotic convergence of the method.

Interior penalty functions have the advantage of approaching the optimum from the feasible region thus yielding a feasible solution. However, the penalty function is discontinuous at the constraint boundaries. Also, the same problems of ill-conditioning and slow convergence exist as seen by Example 2.

Example 2. Solve Example 1 by the interior penalty function method.

Solution: One form of the interior penalty function is

$$F_{\text{int}}(x) = f(x) - \frac{1}{c} \sum_{i=1}^m \frac{1}{g_i(x)} \quad (2.11)$$

Substituting,

$$F_{\text{int}}(x) = -\frac{1}{c} \frac{1}{1-x} \quad (2.12)$$

It can be shown analytically that a minimum exists at

$$x' = 1 + \frac{1}{c}$$

It is again apparent that $x' \rightarrow x^*$ only as $c \rightarrow \infty$.

Figures 3 and 4 illustrate that the same problems of ill-conditioning and slow convergence exist as with the exterior

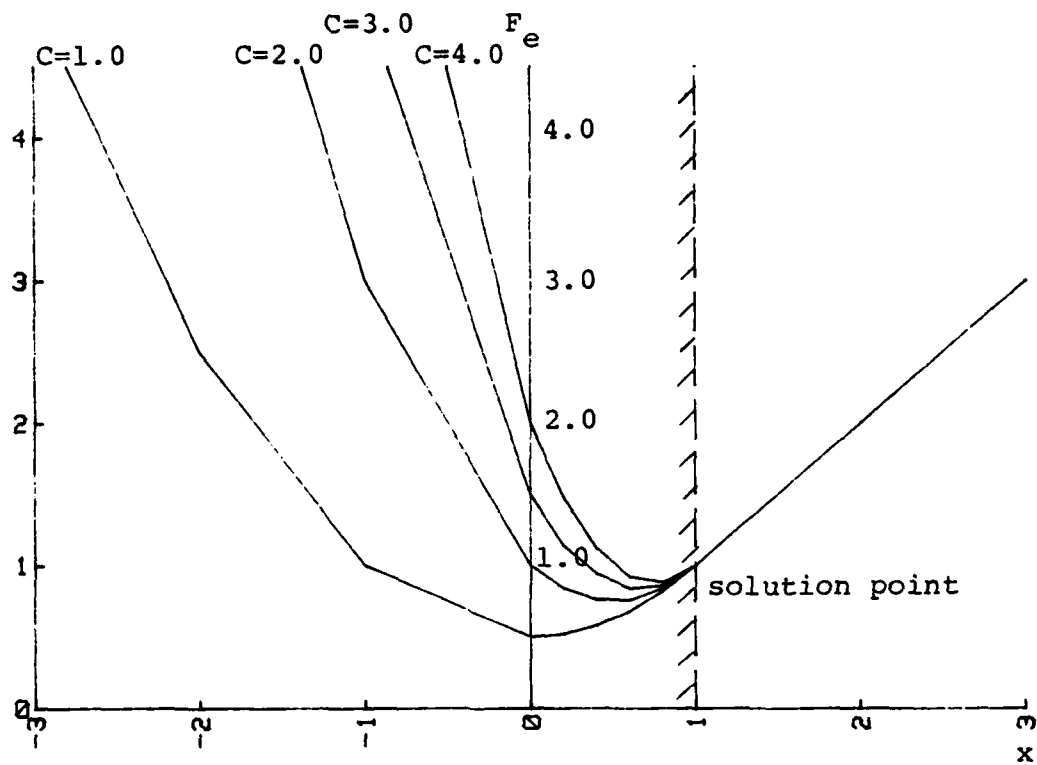


Fig. 1. Exterior Quadratic Penalty Function

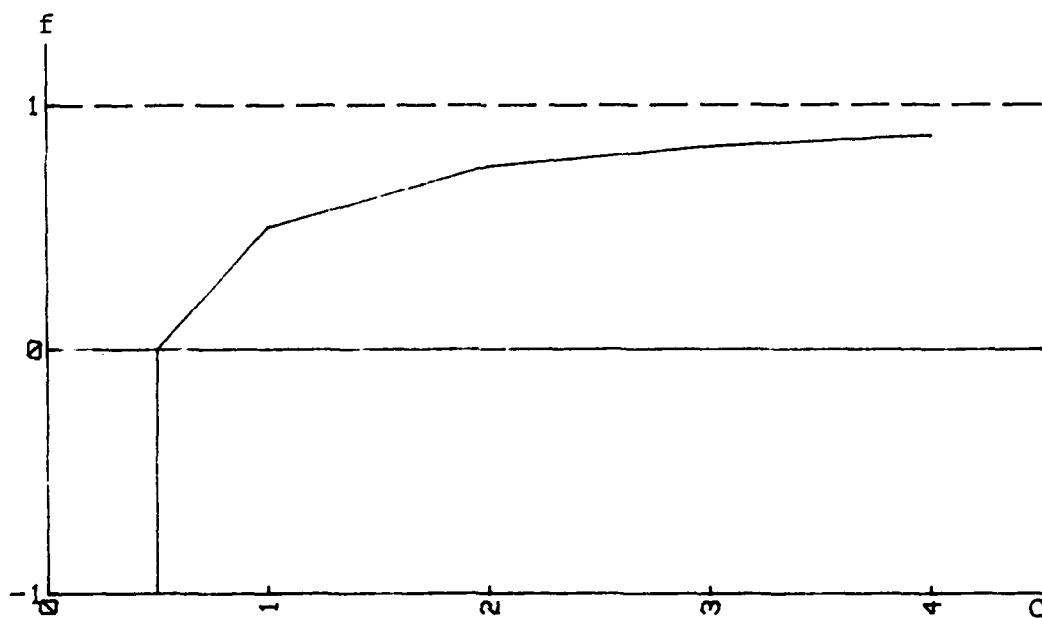


Fig. 2. Convergence of Exterior Penalty Function

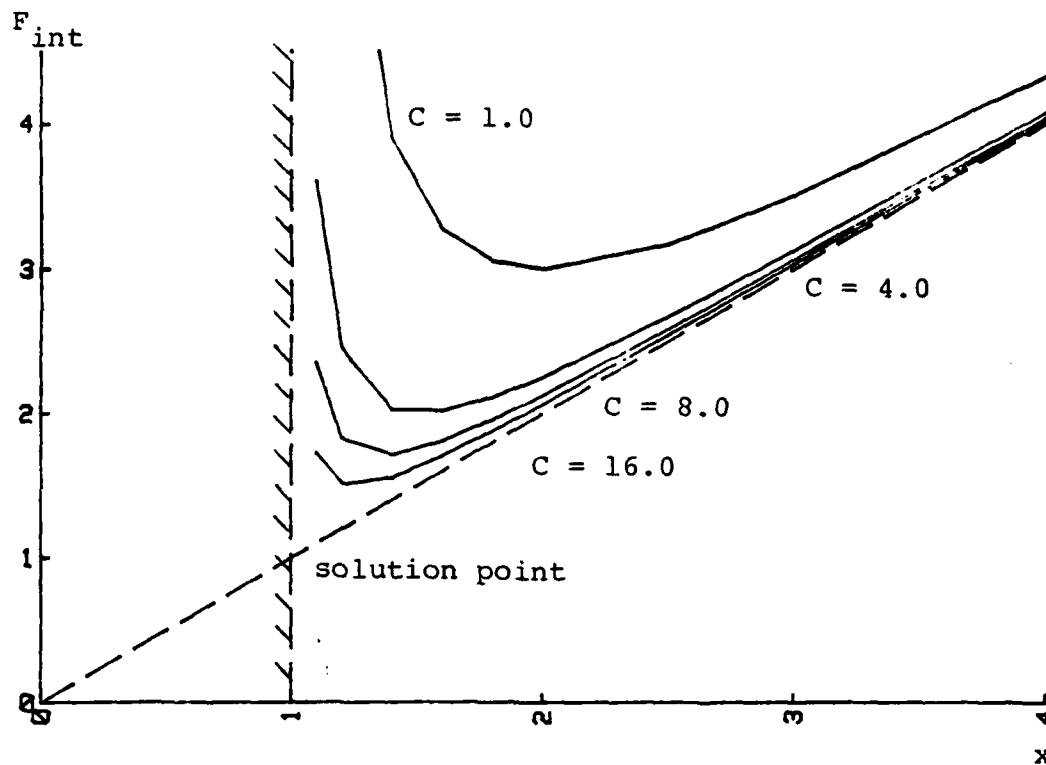


Fig. 3. Interior Penalty Function

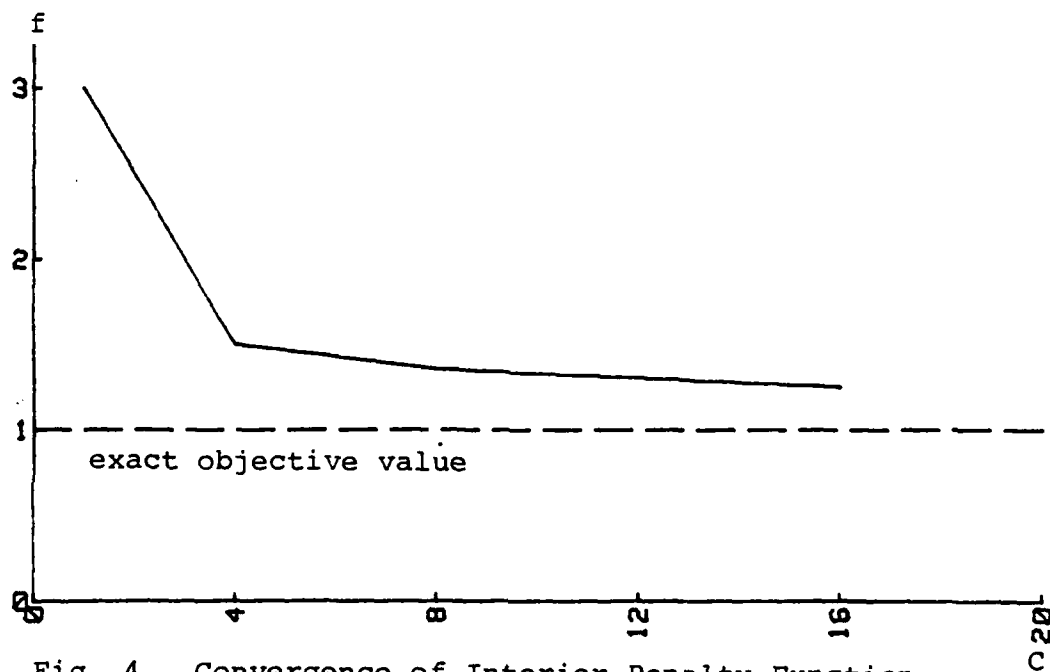


Fig. 4. Convergence of Interior Penalty Function

penalty method. An exact solution is again not possible but the solution achieved here is, however, feasible.

Extended interior penalty methods [19] avoid function discontinuity at $g_i(\bar{X}) = 0$ inherent in the interior penalty method. It is, therefore, less susceptible to ill-conditioning. It does exhibit the same slow convergence of other penalty methods due to the requirement to increase c to infinity. It is for these reasons that the multiplier method is an attractive alternative.

D. THE EQUALITY CONSTRAINED PROBLEM

The multiplier method can be perceived to be a combined primal-dual and penalty function methods. Though they are theoretically similar, their behavior is quite different.

It has been shown that the original equality constrained problem (Eq. 2.1) is equivalent to the classical Lagrangian (Eq. 2.3). Since Equation 2.3 is still an equality constrained problem, it can be solved by the usual exterior penalty function method. The quadratic penalty function is used so first derivatives are continuous. Substituting Equation 2.3 into Equation 2.8:

$$\begin{aligned} A(\bar{X}, \bar{\lambda}, c) &= L(\bar{X}, \bar{\lambda}) + \frac{c}{2} \sum_{i=1}^m p_i^2(\bar{x}) \\ &= f(\bar{X}) + \sum_{i=1}^m \lambda_i h_i(\bar{X}) + \frac{c}{2} \sum_{i=1}^m p_i^2(\bar{X}) \end{aligned} \quad (2.13)$$

where $p_i(\bar{X}) = h_i(\bar{X})$

Equation 2.13 is defined to be the "Augmented Lagrange function" for the equality constrained problem.

By the nature of penalty function methods, as c goes to infinity, Equation 2.13 converges to the solution of Equation 2.8. Concurrently, by choosing an appropriate value of $\bar{\lambda}$ with a suitable update formula, Equation 2.13 can be solved by a series of unconstrained minimizations to obtain a solution to the original problem (Eq. 2.1).

The selection of $\bar{\lambda}$ can be significant to the behavior of the function as seen in the following two extreme cases. First, take $\lambda_i = 0$ for all unconstrained minimizations.

Equation 2.13 becomes

$$A(\bar{X}, \bar{\lambda}, c) = f(\bar{X}) + \frac{c}{2} \sum_{i=1}^m h_i^2(\bar{X}) \quad (2.14)$$

which is the usual quadratic exterior penalty function (Eq. 2.8). It has previously been shown that the function only converges to a minimum $f(\bar{X}^*)$ as c goes to infinity. It has also been shown to be slow in converging, susceptible to numerical ill-conditioning, and to attain only a near optimum solution from the infeasible region.

Next, consider the case where $\lambda_i^0 = \lambda_i^*$. At the minimum, the stationary condition requires that

$$\begin{aligned}\nabla A(\bar{X}^*, \bar{\lambda}^*, c) &= \nabla f(\bar{X}^*) + \sum_{i=1}^m \lambda_i^* h_i(\bar{X}^*) \\ &= \epsilon \sum_{i=1}^m h_i(\bar{X}^*) \nabla h_i(\bar{X}^*) = 0\end{aligned}\quad (2.15)$$

The feasibility condition $h_i(\bar{X}^*)=0$ implies that Equation 2.13 is independent of the value of c . This leads to two important results. First, if the optimum $\bar{\lambda}^*$ is known initially, the solution can be obtained in one unconstrained minimization. Second, since $A(\bar{X}, \bar{\lambda}, c)$ is independent of c at the optimum, it is not necessary to sequentially increase c to infinity to attain a solution. The second result implies that a finite c can be chosen, thus avoiding the inherent ill-conditioning of the penalty methods.

The task is now to find initial values for c and $\bar{\lambda}$ with sequential update formulas for each to achieve a suitable rate of convergence. Collecting terms, Equation 2.15 becomes

$$\nabla f(\bar{X}^*) + \sum_{i=1}^m [\lambda_i^* + c h_i(\bar{X}^*)] \nabla h_i(\bar{X}^*) = 0 \quad (2.16)$$

By the Kuhn-Tucker conditions, at the optimum $(\bar{X}^*, \bar{\lambda}^*)$.

$$\nabla f(\bar{X}^*) + \sum_{i=1}^m \lambda_i^* \nabla h_i(\bar{X}^*) = 0 \quad (2.17)$$

Thus, Equation 2.16 reduces to Equation 2.17 in the limit.

This implies as update formula for $\bar{\lambda}$ such that

$$\lambda_i^{k+1} = \lambda_i^k + c h_i(\bar{X}^k) \quad (2.18)$$

where \bar{X}^k is the solution to the k^{th} unconstrained minimization. Proposed initially by Hestenes [6], Equation 2.18 remains the most popular update formula for $\bar{\lambda}$.

Little experimentation has been done with choosing and initial $\bar{\lambda}^0$. As shown earlier, if $\bar{\lambda}^0 = \bar{\lambda}^*$ the solution is obtained in one unconstrained minimization. Obviously, the closer $\bar{\lambda}^0$ is to $\bar{\lambda}^*$, the more rapid the convergence. A widely accepted practice is to choose $\lambda_i^0 = 0$ due to the computational convenience, and because no other multiplier has consistently proven more efficient. Active and violated constraints are immediately identified in this case since a non-zero λ_i can only occur for an active or violated constraint. A constraint is active if the multiplier is at its optimum $\lambda_i^* \neq 0$ when $\bar{X} \rightarrow \bar{X}^*$. This eliminates any extra computations to check constraint behavior.

Before determining the choice of the initial penalty parameter c^0 and the update formula for c , it is first convenient to examine the convergence of the method which is directly related to the choice of c . Various proofs of linear convergence to a local minimum have been developed [8,20,21]. Rockafellar [13] proved global convergence of the method for convex programming. Bertsekas [14,15] provides a rigorous proof of the method's convergence for the general nonlinear problem. He compares the convergence of the various multiplier methods, penalty methods, and primal-dual methods [22]. His results for the Hestenes' [6] multiplier method are summarized here.

Recall the assumptions made for a local minimum to exist:

Assumption 1. There exists a local minimizing point \bar{X}^k of problem 2.1 which satisfies the second

order sufficiency conditions for an isolated local minimum, i.e., f and h_i are twice continuously differentiable in a neighborhood of \bar{X}^k , the gradients $\Delta h_i(\bar{X}^k)$, ($i=1, \dots, m$) are linearly independent and there exists a Lagrange multiplier vector $\bar{\lambda}$, such that $\nabla L(\bar{X}^k, \bar{\lambda}^k) = 0$ and $|\nabla^2 L(\bar{X}^k, \bar{\lambda}^k)| > 0$ [22].

Assumption 2. The penalty function $\phi(t)$ is twice continuously differentiable in an open interval containing zero and $\phi''(0) > 0$, where ϕ'' denotes the second derivative [22].

Also, assuming the Hessian matrices $\nabla^2 f(\bar{X})$, $\nabla^2 h_i(\bar{X})$, and the second derivative ϕ'' are Lipschitz continuous,¹ there exists a scalar $c^* > 0$, and $M > 0$ such that for every $c^k > c^*$ the function $L_c(\bar{X}, \bar{\lambda})$ has a unique minimum $\bar{X}^k(\bar{\lambda}, c)$. Furthermore,

$$||\bar{X}^k - \bar{X}^*|| \leq \frac{M ||\bar{\lambda}^k - \bar{\lambda}^*||}{c} \quad (2.19)$$

$$\text{and} \quad ||\bar{\lambda}^{k+1} - \bar{\lambda}^*|| \leq \frac{M ||\bar{\lambda}^k - \bar{\lambda}^*||}{c} \quad (2.20)$$

$$\text{where} \quad \lambda_i^{k+1} = \lambda_i^k + c h_i(\bar{X}^k)$$

The notation $||\cdot||$ denotes the usual Euclidean norm.

From Equation 2.20, if $c^k \rightarrow c < \infty$ for the non-decreasing positive penalty sequence $\{c^k\}$ so as to ensure $M/c^k \rightarrow 1$, then the sequence $\{\bar{\lambda}^k\}$ converges to $\bar{\lambda}^*$ linearly. If c^k goes to infinity,

¹A function is said to satisfy a Lipschitz condition of order m on the closed interval (a, b) if there exists a constant c such that $|f(x_2) - f(x_1)| < c|x_2 - x_1|^m$ for all values x_1, x_2 on (a, b) .

$M/c^k \rightarrow 0$, and the sequence $\{\bar{\lambda}^k\}$ converges superlinearly. From Equation 2.19, for sufficiently large but finite c^k , the sequence $\{\bar{X}^k\}$ converges to X^* since $\{\bar{\lambda}^k\}$ converges. If c^k goes to infinity, $\{\bar{X}^k\}$ converges by the same argument.

The rate of convergence of $\{\bar{X}^k\}$ is represented by the right hand side of Equation 2.19. Since $\{\bar{\lambda}^k\}$ converges at least linearly, $\{\bar{X}^k\}$ converges at least linearly. This has been shown to be significantly faster than pure penalty or primal-dual methods. Numerical ill-conditioning is avoided since this linear convergence rate is achieved with a sufficiently large finite c .

The convergence criterion of Equations 2.19 and 2.20 are global in nature since no bounds restriction has been imposed. No restrictions on the convex or non-convex nature of the problem are specified as well. The global convergence of the multiplier method is contingent upon the ability of the unconstrained minimization method to generate a sequence $\{\bar{X}^k\}$ which are well defined local minimums to the function $A(\bar{X}^k, \bar{\lambda}^k, c)$. Naturally, the function $A(\bar{X}^k, \bar{\lambda}^k, c)$ may have other local minimums to which the unconstrained minimization method may be attracted. Unless the unconstrained minimization method stays in the neighborhood of the same local minimum, the convergence argument is invalid and there is no assurance that the multiplier method will perform any better or worse than the penalty methods. It should be noted that the usual practice is to use the last point \bar{X}^k of the k^{th} minimization

as the starting point of the $(k+1)$ th minimization. This generally produces sequences $\{\bar{x}^k\}$ which are close to the same local minimum of $A(\bar{x}, \bar{\lambda}, c)$.

The convergence rate of the multiplier iteration (Eq. 2.18) is linear with the convergence ratio essentially inversely proportional to the penalty parameter c [22]. This fact is strongly dependent on Assumptions 1 and 2. If either of the assumptions is relaxed, the convergence rate may become sublinear or superlinear as the following examples show.

Example 3 [22]. Consider the scalar problem minimize $x^2/2$ such that $x=0$ with an optimal solution $x^*=0$, $\lambda^*=0$. In this example Assumption 2 is not satisfied. For $\lambda \leq 0$, $\phi(t)=|t|^3/3$, and $c=1$, Equation 2.13 becomes

$$x^1(\lambda, 1) = \frac{-1 - \sqrt{(1-4\lambda)}}{2}$$

Starting at $\lambda^0=0$, Equation 2.18 becomes

$$\lambda^{k+1} = \frac{1 - \sqrt{(1-4\lambda^k)}}{2}$$

The $\lim_{k \rightarrow \infty} \left(\frac{\lambda^{k+1}}{\lambda^k} \right) = 1$, i.e., a sublinear convergence rate.

Example 4 [22]. Consider Example 3 where $\phi(t)=2\sqrt{|t|}/3$, $c=1$. Again, Assumption 2 is not satisfied. The solution to Equation 2.13 becomes

$$x^1(\lambda, 1) = \frac{(-1 + \sqrt{(1-4\lambda)})^2}{4}$$

Starting at $\lambda^0=0$, Equation 2.18 becomes

$$\lambda^{k+1} = \lambda^k + \frac{(-1 + \sqrt{1-4\lambda^k})}{2}$$

It can be shown that

$$\lim_{k \rightarrow \infty} \frac{|\lambda^{k+1}|}{(\lambda^k)^2} = 1$$

hence, a superlinear convergence (order 2).

Example 5 [22]. Consider the problem, minimize $|x|^{3/2}$ such that $x=0$. Again $x^*=\lambda^*=0$, but Assumption 1 is not satisfied. For $\phi(t)=t^2/2$, $c=1$, the solution to Equation 2.13 becomes

$$x^k(\lambda, 1) = \frac{(-1 + \sqrt{1-4\lambda})}{2}$$

and Equation 2.18 becomes

$$\lambda^{k+1} = \lambda^k + \frac{(-1 + \sqrt{1-4\lambda^k})}{2}$$

Again,

$$\lim_{k \rightarrow \infty} \frac{|\lambda^{k+1}|}{(\lambda^k)^2} = 1$$

hence, superlinear convergence.

It can be shown that the convergence rate is influenced substantially by the rates of change of the derivatives (curvature) of the primal function $p(u) = \min f(\bar{X})$, where $u = h(\bar{X})$, and the penalized primal function derivatives $p(u)+c[\phi(u)]$ near $u=0$. The convergence is faster if the rate of change of $\nabla p(u)$ is small and the rate of change of $c[\nabla \phi(u)]$ is large near $u=0$.

In Example 3, the rate of change of $\nabla\phi(u)$ is small near $u=0$ and convergence is slow, while in Example 4, it is large and convergence is rapid. In Example 5, the rate of change of $\nabla p(u)$ is small near $u=0$, thus the fast convergence.

The following example shows that in the absence of Assumption 1, Equation 2.18 may not lead to convergence for any $c>0$ when ϕ is essentially quadratic.

Example 6 [22]. Consider the problem, minimize $-|x|^\rho$ such that $x=0$ where $1<\rho<2$. For any $c>0$, there exists a neighborhood of $x=0$ such that the augmented Lagrangian (Eq. 2.14) does not have a local minimum for any λ when ϕ is essentially quadratic. This can be corrected by using $\phi(t)=|t|^{\rho'} + t^2/2$ where $1<\rho'<\rho$.

The two extreme cases are now examined. First, suppose $\bar{\lambda}^k = \bar{\lambda}^*$, Equation 2.19 becomes

$$|| \bar{X}^k - \bar{X}^* || \leq 0$$

The norm is always non-negative; therefore, $\bar{X}^k = \bar{X}^*$ and the solution is reached in one unconstrained minimization.

Next, letting $\bar{\lambda}^k = 0$ for all k , i.e., the penalty function method, Equation 2.19 becomes

$$|| \bar{X}^k - \bar{X}^* || < \frac{M || \bar{\lambda}^* ||}{c}$$

In this case, $\bar{X}^k \rightarrow \bar{X}^*$ only if $c \rightarrow \infty$, requiring many unconstrained minimizations and a sublinear rate of convergence.

It has been shown that the rate of convergence is directly dependent on the penalty parameter, c . The convergence estimates (Eqs. 2.19, 2.20) are valid for c greater than some

threshold value, c^* which depends on $(\bar{\lambda})$ and the problem data. In general, c^* is unavailable and it is impossible to know apriori the range of values of c for which Equations 2.19 and 2.20 are valid and imply fast convergence.

A penalty parameter update sequence is required to increase c^k monotonically with each unconstrained minimization. As $c \rightarrow \infty$, Equations 2.19 and 2.20 will eventually become valid. It should be noted that large values of c can induce ill-conditioning, making the unconstrained minimization of $A(\bar{X}, \bar{\lambda}, c)$ difficult. On the other hand, Equations 2.19 and 2.20 indicate faster convergence of $\{\bar{\lambda}^k\}$ to $\bar{\lambda}^*$ for large values of c .

An update sequence is recommended whereby c is multiplied by some constant $\gamma > 1$, i.e., $c^{k+1} = \gamma c^k$. The penalty parameter c is increased in this manner to some significantly large c_{\max} . Bertsekas [22] recommends γ not much larger than 1 to avoid ill-conditioning effects in the first few unconstrained minimization iterations. This update scheme will be used since it has been subject to the most testing in recent years. The choice of an initial c requires more experimentation and will be discussed later.

Other methods are available for updating the penalty parameter c . Powell [7] suggests multiplying c by some constant $\beta > 1$ only if the violated constraint, as measured by $||h[\bar{X}(\bar{\lambda}, c)]||$, is not decreased by a certain factor over the previous minimization, i.e., $c^{k+1} = \beta c^k$ if $||h[\bar{X}(\bar{\lambda}^k, c^k)]|| > \gamma ||h[\bar{X}(\bar{\lambda}^{k-1}, c^{k-1})]||$ and $c^{k-1} = c^k$ otherwise, where $\beta > 1$, and, $\gamma < 1$ are some specified

scalars. This scheme generates a penalty parameter sequence that will be constant after a certain index and will achieve convergence by virtue of enforcing the asymptotic feasibility of the constraints, i.e.,

$$\lim_{k \rightarrow \infty} ||h[\bar{X}(\bar{\lambda}^k, c^k)]|| = 0$$

Another similar update scheme is to use a penalty parameter vector such that c_i is updated by the Powell method only if the constraint $h_i[\bar{X}(\bar{\lambda}, c)]$ is violated. This case of a separate penalty factor for each constraint corresponds to merely scaling of the constraints. A simple modification to Equations 2.19 and 2.20 is used to prove convergence of this case.

Finally, define a dual function $A_c(\bar{\lambda})$ for the augmented Lagrange function $A(\bar{X}, \bar{\lambda}, c)$ such that

$$A_c(\bar{\lambda}) = \min_{\bar{X}} A(\bar{X}, \bar{\lambda}, c) \quad (2.21)$$

It can be shown that if $\bar{X}^k(\bar{\lambda}, c)$ is the solution to Equation 2.21, then $A_c(\bar{\lambda})$ is a twice continuously differentiable convex function with gradient given by

$$\nabla_{\lambda} A_c(\bar{\lambda}) = \nabla h[\bar{X}^k(\bar{\lambda}, c)] \quad (2.22)$$

where $h[\bar{X}^k(\bar{\lambda}, c)]$ is the constraint vector (Eq. 2.1b) evaluated at \bar{X}^k . Substituting Equation 2.22, Equation 2.18 becomes

$$\bar{\lambda}^{k+1} = \bar{\lambda}^k + c^k \bar{\nabla}_{\lambda} A_c(\bar{\lambda}) \quad (2.23)$$

This relation shows that the multiplier iteration (Eq. 2.18) is an iteration of steepest ascent for finding the maximum of the dual function, $A_c(\bar{\lambda})$. Equation 2.23 is equivalent to

a steepest ascent iteration for quadratic penalty functions, $\phi(t) = t^2/2$. The multiplier method can thus be viewed as a primal-dual method with a limited search for the optimal Lagrange multipliers in the dual space.

The Hestenes' multiplier method for nonlinear equality constrained problems will not be summarized. Quadratic penalty functions are used in this algorithm.

- Step 1: Select $\bar{\lambda}^0 = 0$ and an appropriate penalty parameter $c^0 > 0$. Set $k = 1$.
- Step 2: Solve $\min A(\bar{X}, \bar{\lambda}^k, c)$, defined by Equation 2.13. Denote the solution \bar{X}^k .
- Step 3: Update $\bar{\lambda}^k$ by Equation 2.18.
- Step 4: If $\bar{\lambda}^{k+1} = \bar{\lambda}^k$, stop. $(\bar{X}^k, \bar{\lambda}^k)$ is the optimal solution. Otherwise, go to Step 5.
- Step 5: Set $\bar{\lambda}^k = \bar{\lambda}^{k+1}$. If $c^k < c_{\max}$, where c_{\max} is some significantly large number, update c^k by $c^{k+1} = \gamma c^k$, where γ is some increase factor greater than one. Otherwise, $c^k = c_{\max}$. Set $k = k + 1$. Go to Step 2.

Note that no assumptions or restrictions have been made concerning the nature of the objective or constraint function. If Assumptions 1 and 2 are satisfied, the solution will converge to a global minimum; otherwise, only a local minimum can be guaranteed.

E. THE INEQUALITY CONSTRAINED PROBLEM

The multiplier method has been shown to be theoretically attractive for equality constrained minimization problems. In reality, many engineering problems involve inequality constraints. It is, therefore, necessary to extend the discussion of the multiplier method to include inequality constraints.

Consider the nonlinear inequality constrained problem:

$$\text{Minimize } f(\bar{X}) \quad (2.24a)$$

$$\text{such that } g_i(\bar{X}) \leq 0 \quad (i=1, \dots, \ell) \quad (2.24b)$$

where \bar{X} is the vector of n design variables. Introducing slack variables, Equation 2.24b becomes

$$g_i(\bar{X}) + z_i^2 = 0 \quad (2.25)$$

where z_i^2 is the slack variable for the i^{th} constraint. The problem is now an equality constrained problem of the form of Equation 2.1; however, the number of design variables has increased to $n+\ell$. The augmented Lagrangian (Eq. 2.13) becomes

$$\begin{aligned} A(\bar{X}, \bar{z}, \bar{\lambda}, c) = & f(\bar{X}) + \sum_{i=1}^{\ell} \lambda_i (g_i(\bar{X}) + z_i^2) \\ & + \frac{c}{2} \sum_{i=1}^{\ell} (g_i(\bar{X}) + z_i^2)^2 \end{aligned} \quad (2.26)$$

If the number of constraints, ℓ is much greater than the number of design variables, n , as is often the case in engineering design problems, the unconstrained minimization problem becomes sizable. The scope of the problem can, however, be reduced by eliminating the slack variables, z_i^2 .

The unconstrained function, $A(\bar{X}, \bar{Z}, \bar{\lambda}, c)$ is first minimized with respect to \bar{Z} . For a local minimum to exist, the stationary conditions

$$\frac{\partial A}{\partial z_i} = 0 \quad (i=1, \dots, n) \quad (2.27)$$

must hold. Differentiating $A(\bar{X}, \bar{Z}, \bar{\lambda}, c)$

$$\frac{\partial A}{\partial z_i} = 2\lambda_i x_i + c[g_i(\bar{X}) + z_i^2] (2z_i) = 0 \quad (2.28)$$

$$z_i \left[\frac{\lambda_i}{c} + g_i(\bar{X}) + z_i^2 \right] = 0 \quad (2.29)$$

The solution to Equation 2.29 is

$$z_i^2 = 0$$

or

$$z_i^2 = -\frac{\lambda_i}{c} - g_i(\bar{X})$$

Since $z_i^2 < 0$ is meaningless, the solution becomes

$$z_i^2 = \max \left[0, -g_i(\bar{X}) - \frac{\lambda_i}{c} \right] \quad (2.30)$$

Equation 2.30 shows that z_i is no longer an independent variable. From this equation it is observed that if $g_i(\bar{X})$ is a critical constraint, $z_i = 0$. If $g_i(\bar{X})$ is non-critical, $z_i > 0$. Therefore,

$$g_i(\bar{X}) + z_i^2 = \max \left[g_i(\bar{X}), -\frac{\lambda_i}{c} \right] \quad (2.31)$$

With the slack variables eliminated, the augmented Lagrangian becomes

$$A(\bar{X}, \bar{\lambda}, c) = f(\bar{X}) + \sum_{i=1}^{\ell} \left[\lambda_i \psi_i + \frac{c}{2} \psi_i^2 \right]$$

$$\text{where } \psi_i = \max \left[g_i(\bar{X}), -\frac{\lambda_i}{c} \right] \quad (2.32)$$

This is referred to as Rockafellar's augmented Lagrange function. Note that $A(\bar{X}, \bar{\lambda}, c)$ has continuous first derivatives with respect to c , but discontinuous second derivatives at $g_i(\bar{X}) = -\bar{\lambda}/c$. This precludes the use of second order methods, i.e., Newton's method, for unconstrained minimization.

From Equation 2.31, Hestenes' update formula for $\bar{\lambda}$ for the inequality constrained problem becomes

$$\lambda_i^{k+1} = \max [0, \lambda_i^k + c g_i(\bar{X}^k)] \quad (2.33)$$

The algorithm for Hestenes' multiplier method is easily modified to include inequality constraints by adding Equations 2.32 and 2.33 to steps 2 and 3, respectively. Note that since the inequality constrained problem is transformed to an equivalent equality constrained problem, the convergence properties are identical.

Example 1 can now be solved by the multiplier method.

Example 7 [2]. Minimize x such that $1-x \leq 0$.

Solution: Equation 2.32 becomes

$$A(x, \lambda, c) = x + c \begin{cases} \frac{1}{2}(1-x)^2 + \frac{\lambda}{c}(1-x), & \text{if } 1-x \geq 0 \\ -\frac{\lambda^2}{2c}, & \text{otherwise} \end{cases}$$

Applying Hestenes' algorithm with $\lambda^0=0$ and $c^0=1$, the augmented Lagrangian becomes

$$A^0(x, 0, 1) = x + \frac{1}{2} \begin{cases} (1-x)^2, & \text{if } 1-x \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

This has a solution $x^0=0$. Updating λ by Equation 2.32,

$$\begin{aligned}\lambda^1 &= \max [0, \lambda^0 + cg(x^1)] \\ &= \max [0, 0 + 1(1)] = 1\end{aligned}$$

With $\lambda^1=1$ and $c^1=1$, the next unconstrained minimization of $A(x, \lambda, c)$ becomes

$$A^1(x, 1, 1) = x + \begin{cases} (1-x)^2/2 + (1-x), & \text{if } 1-x \geq 0 \\ -1/2, & \text{otherwise} \end{cases}$$

The solution becomes $x' = 1$. Updating λ ,

$$\begin{aligned}\lambda^2 &= \max [0, \lambda^1 + c^1 g^1(x^1)] \\ &= \max [0, 1 + 1(0)] = 1\end{aligned}$$

Since $\lambda^1 = \lambda^2$, stop the calculation. The optimal solution is $(x^*, \lambda^*) = (1, 1)$.

Note the convergence of the method even for a constant penalty parameter c .

The functions A^0 and A^1 are plotted in Fig. 5, and the objective value vs. c is plotted in Fig. 6. Several facts are evident from this example.

1. A solution is obtained in a few unconstrained minimizations.
2. Each unconstrained problem is a smooth curve. Thus ill-conditioning is avoided.
3. Convergence is not asymptotic and an exact solution is attainable.
4. With an initial Lagrange multiplier of zero, the solution is obtained from the infeasible region.

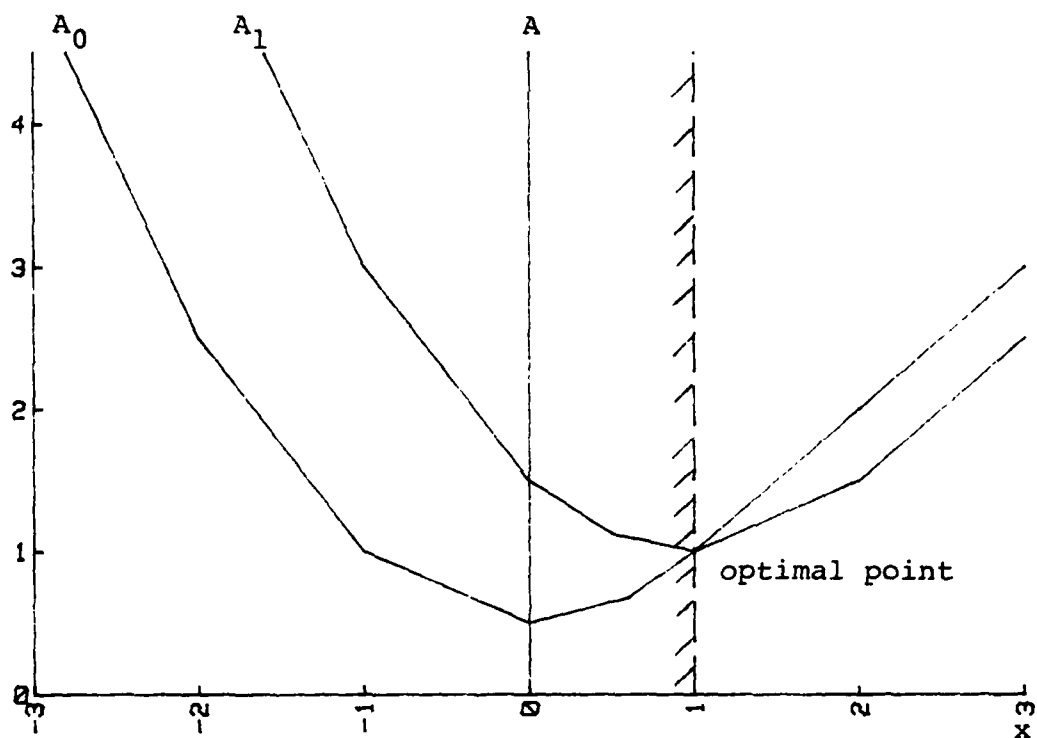


Fig. 5. Augmented Lagrangian Function

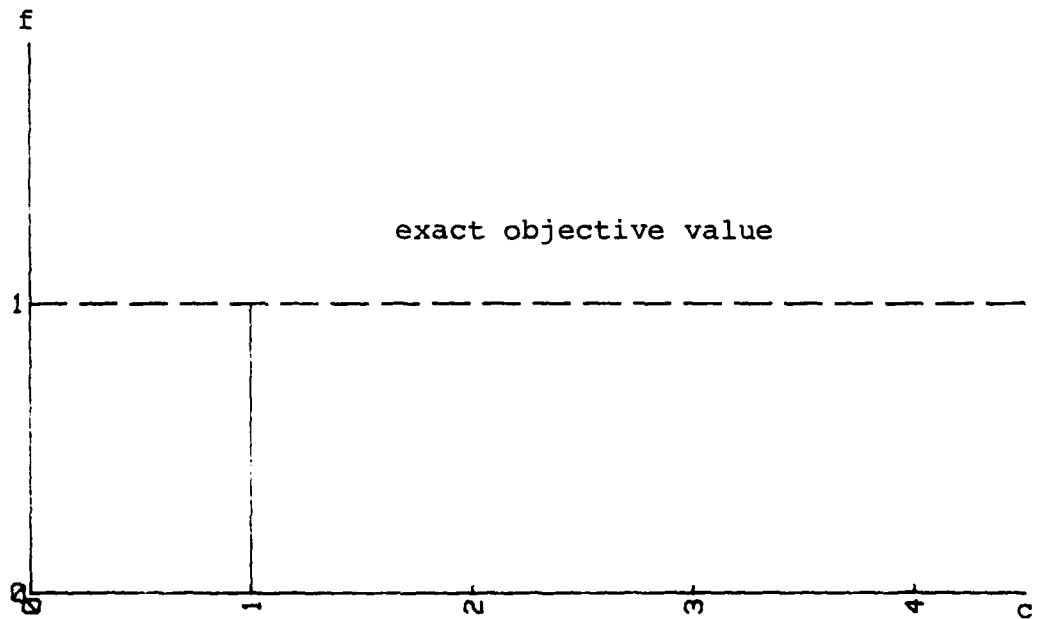


Fig. 6. Convergence of Augmented Lagrangian Function

5. The optimal Lagrange multiplier is obtained.

6. Any starting point is possible.

The method, therefore, shows attractive theoretical features over pure penalty or primal-dual methods.

The multiplier method algorithm is now summarized for the nonlinear equality and inequality constrained problem:

Minimize $f(\bar{X})$

such that $g_i(\bar{X}) \leq 0 \quad (i=1, \dots, \ell)$

and $h_j(\bar{X}) = 0 \quad (j=\ell+1, \dots, \ell+m)$

Algorithm:

Step 1. Choose an initial Lagrange multiplier $\bar{\lambda}^0 = 0$ and an initial penalty parameter $c^0 > 0$.

Step 2. Solve the unconstrained minimization problem

$$\text{Min } A(\bar{X}, \bar{\lambda}, c) = f(\bar{X})$$

$$+ \sum_{i=1}^m [\lambda_i \psi_i + \frac{c}{2} \psi_i^2]$$

$$+ \sum_{j=\ell+1}^{\ell+m} [\lambda_j h_j(\bar{X}) + \frac{c}{2} h_j^2(\bar{X})]$$

$$\text{where } \psi_i = \max [g_i(\bar{X}), -\frac{\lambda_i}{c}]$$

Step 3. Update the Lagrange multipliers, $\bar{\lambda}$ by

$$\lambda_i^{k+1} = \lambda_i^k + c \max[g_i(\bar{X}^k), -\frac{\lambda_i^k}{c}] \quad (i=1, \dots, \ell)$$

$$\lambda_j^{k+1} = \lambda_j^k + c h_j(\bar{X}^k) \quad (j=m+1, \dots, m+\ell)$$

Step 4. If $\bar{\lambda}^k = \bar{\lambda}^{k+1}$, stop. The optimal solution is $(\bar{X}^*, \bar{\lambda}^*) = (\bar{X}^k, \bar{\lambda}^k)$. Otherwise, go to Step 5.

Step 5. Set $\bar{\lambda}^k = \bar{\lambda}^{k+1}$. Update the penalty parameter c by $c^{k+1} = \gamma c^k$.

If $c^{k+1} \geq c_{\max}$, set $c^k = c_{\max}$, else set $c^k = c^{k+1}$.

Set $k = k+1$; go to Step 2.

This is the theoretical algorithm used in the multiplier method. Minor variations are described in the next chapter which are designed to improve the computational efficiency, reliability, and accuracy of the method.

III. COMPUTATIONAL ASPECTS

The effectiveness of the multiplier method is dependent upon the accuracy of the unconstrained minimization, choice of Lagrange multiplier update formula, penalty parameter, the form of the penalty function, and the objective and constraint functions themselves. The theoretical effect of the Lagrange multiplier and penalty parameters on the method's convergence rate was described in Chapter II. The requirement that the objective and the constraint functions satisfy Assumptions 1 and 2, listed in Chapter II.D, has also been established.

This chapter will examine the effects of the Lagrange multiplier and penalty parameter from a computational viewpoint. As stated earlier, the penalty function is quadratic throughout this discussion. The computational aspects of the unconstrained minimization program and associated one-dimensional search subprograms are first examined.

A. THE UNCONSTRAINED MINIMIZATION PROBLEM

A sequence of unconstrained minimizations is required in obtaining a solution by the multiplier method. Of the many unconstrained minimization techniques available, the Davidon-Fletcher-Powell [DFP] method [1] was selected because of its accuracy and rapid convergence. Figure 7 compares the DFP

COMPARISON OF UNCONSTRAINED MINIMIZATION METHODS

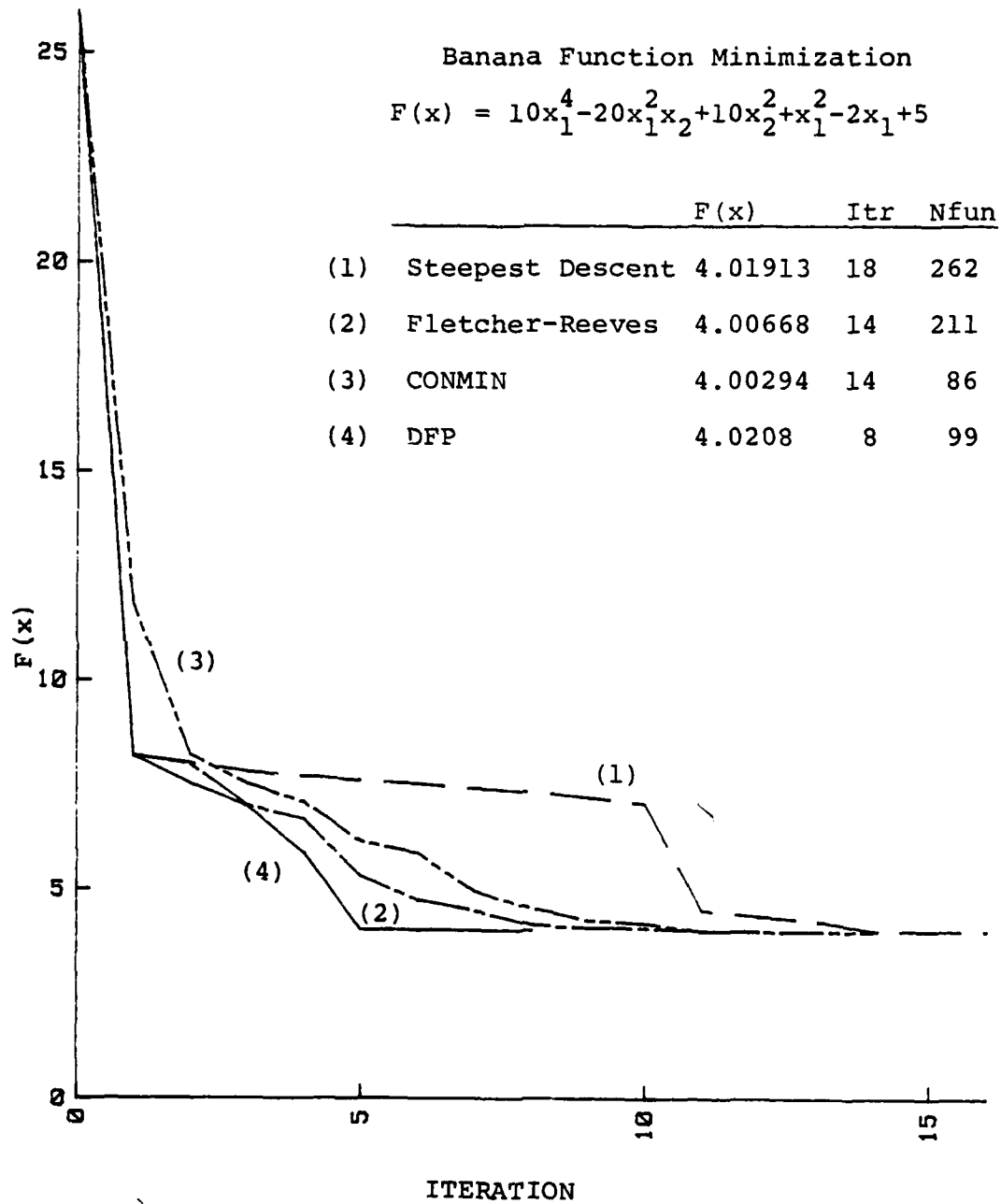


Fig. 7. Comparison of Unconstrained Minimization Methods

method to the methods of steepest descent, Fletcher-Reeves conjugate gradient method developed as part of this study, and the Fletcher-Reeves method contained in the CONMIN program for the extremely ill-conditioned "banana" function. Note the DFP method converges in fewer iterations than the other methods with little or no loss of accuracy.

To solve the general unconstrained minimization problem, it is necessary to first calculate a search direction, \bar{S} , initially the direction of steepest descent. A one-dimensional search is then used to find the minimum of the objective function in the search direction, \bar{S} . The one-dimensional search computes a scalar, α_k^* which minimizes the function $f(\bar{X}_{k-1} + \alpha_k \bar{S}_k)$. The new X-vector associated with the function minimum becomes

$$\bar{X}_k = \bar{X}_{k-1} + \alpha_k \bar{S}_k$$

A new search direction is calculated, and the iteration is repeated until a specified convergence criteria is satisfied.

The algorithm for the DFP method is described as follows for the general unconstrained minimization problem:

- Step 1. Select an initial point \bar{X}_0 and an initial positive definite symmetric matrix, $H_0 = I$ (identity matrix). Set $k=1$.
- Step 2. Calculate the gradient, $\nabla f(\bar{X}_0)$, and set

$$\bar{S}_0 = - H_0 \nabla f(\bar{X}_0)$$

- Step 3. Compute $\bar{X}_k = \bar{X}_{k-1} + \alpha_k^* \bar{S}_k$

where α_k^* minimizes $f(\bar{X}_{k-1} + \alpha \bar{S}_k)$, i.e., α_k^* is the solution to the one-dimensional search.

Step 4. Computer $H_{k+1} = H_k + \bar{A}_k - \bar{B}_k$

where $\Delta \bar{F} = \nabla f(\bar{X}_k) - \nabla f(\bar{X}_{k-1})$

$$\Delta \bar{X} = \bar{X}_k - \bar{X}_{k-1} = -\alpha^* \bar{S}_k$$

$$\bar{A}_k = \frac{\Delta \bar{X} \Delta \bar{X}^T}{\Delta \bar{X}^T \Delta \bar{F}}$$

$$\bar{B}_k = \frac{(H \Delta \bar{F}) (H \Delta \bar{F})^T}{\Delta \bar{F}^T H \Delta \bar{F}}$$

Step 5: Compute $\bar{S}_{k+1} = -H_{k+1} \nabla f_k$, set $k=k+1$, go to Step 3. The basic algorithm is modified to provide (1) scaling of the variables, (2) appropriate update and resets, and (3) a suitable termination criterion.

A feature is added which normalizes the search vector such that

$$S_{k \text{ NORM}} = \frac{\bar{S}_k}{|\bar{S}_k|_{\max}}$$

For excessively ill-conditioned problems, pre-scaling the variables may also be required.

Resets and updates of the vector, \bar{S} and the matrix, H , are incorporated to prevent method breakdown from round-off error and other instabilities. The positive definiteness of H is preserved in theory only if α_k^* provides a true minimum point, i.e., $\nabla \bar{F}_{k+1}^T \bar{S}_k = 0$. The update of H is, therefore, skipped if $\nabla \bar{F}_{k+1}^T \bar{S}_k > \epsilon$, $H \neq I$, and the update

was not skipped on the previous iteration. A tolerance value, $\epsilon = 0.01$ was computationally found to provide a good update criteria with negligible effect on the convergence side.

Two checks were made on \bar{S} to insure a valid search direction. Note first that if $H=I$, the search direction is that of steepest descent. Now at iteration k , if $\bar{S}_k \cdot \nabla f_k(\bar{X}) > 0$, the search direction will not reduce the objective and a new search direction is found. Second, if at the end of the one-dimensional search $f(\bar{X}_{k+1}) > f(\bar{X}_k)$, the function is obviously not a minimum and a new search direction is found. In each case, the H-matrix is reset to identify, I , and a search in the direction of steepest descent is performed. Finally, to maintain stability of the H-matrix, it is reset to the identify matrix every $NDV+1$ iterations, where NDV is the number of design variables.

The method is terminated when the relative or absolute difference of the objective function is less than 0.001 for two consecutive iterations or when a preassigned maximum number of iterations is exceeded. The somewhat strict convergence criterion was chosen since the performance of the multiplier method is highly dependent on an accurate unconstrained minimization.

The DFP method has been shown to be a reliable first order method with quadratic convergence. It has one

additional feature in that it provides a good approximation to the Hessian matrix,

$$H \approx \frac{\partial^2 f(\bar{X})}{\partial \bar{X}^2}$$

Thus, the method provides many of the advantages of a second order, Newton type method, without the tedious calculation of second derivatives.

Finally, the DFP method requires subprograms to calculate gradients and perform the one-dimensional search. Gradient calculation was done by the first forward finite difference method. While a central difference method would improve the accuracy of the calculations, the first forward finite difference method with a step size of 0.01 has provided acceptable results in experimentation thus far. Discussion of the one-dimensional search subprogram follows in the next section.

B. THE ONE-DIMENSIONAL SEARCH

A combination of the Golden Section method and polynomial interpolation was used for the one-dimensional search. The one-dimensional search calculates a scalar, α_k^* which minimizes the function, $f(\bar{X}_{k-1} + \alpha_k \bar{S}_k)$. It has been shown that the accuracy of computing α_k is critical to the performance of the unconstrained minimization subprogram. The Golden Section method was used because of its reliability and accuracy. The method is, however, less efficient than polynomial interpolation methods in many cases. The Golden Section method

converges linearly while polynomial methods have no guaranteed rate of convergence. It may also require many function evaluations per iteration while polynomial methods require few.

The Golden Section method is based on a Fibonacci search [16] which yields a sequence of intervals of uncertainty whose widths tend to zero faster than any other method. Given upper and lower bounds, x_u and x_l , where x denotes the scalar, α , in Equation 3.1, two interior points, x_1 and x_2 are found such that

$$x_u - x_2 = x_1 - x_l$$

and the ratio

$$\frac{x_2}{x_1} = (1 + \sqrt{5})/2 = 1.61803$$

This is the Golden Section number. The interior points, x_1 and x_2 are defined as

$$x_1 = (1 - \tau)x_l + \tau x_u$$

$$x_2 = \tau x_l + (1 - \tau)x_u$$

$$\text{where } \tau = (3 - \sqrt{5})/2 = 0.38197$$

The bounds are revised by a simple update routine and one new interior point is calculated at each step. The process is continued until a given relative convergence tolerance, ϵ , is satisfied. The algorithm for the Golden Section search is given in Appendix A.

The relative convergence tolerance, ϵ , is defined

$$\epsilon = \Delta x / (x_u - x_l)$$

Recognizing that the interval is reduced by the fraction, τ (38 percent) each iteration, the maximum number of function evaluations, N per iteration is calculated

$$\epsilon = (1-\tau)^{N-3}$$

or

$$N = \ln(\epsilon)/\ln(1-\tau) + 3$$
$$= -2.078 \ln(\epsilon) + 3$$

Note that by defining ϵ , a fixed number of function evaluations are performed each iteration. If the absolute convergence tolerance, Δx were specified, the number of function evaluations will vary each iteration. Experiments have shown an absolute convergence criteria to yield slightly better accuracy with nearly the same number of function evaluations. Therefore, an absolute convergence criteria was specified to give the same or better efficiency than the relative criteria and with improved accuracy.

The Golden Section method requires a logical guess to the first x_u such that x_u and x_l bracket the minimum. Selection of x_u was performed such that if more than two guesses were required, the last three points chosen would yield the values x_l , x_1 , and x_u . This provided a smaller initial bound than if the original x_l was used and also saved one function evaluation since x_1 was already found for future calculations.

Finally, a cubic approximation was performed using the last four points from the Golden Section method. A better

minimum was found in almost every case at the cost of only one additional function evaluation.

C. THE MULTIPLIER METHOD

Experimentation with the multiplier method involved investigating the effects of the penalty parameter, c , the update factor, γ , and the update formula for the multipliers, $\bar{\lambda}$. It was desired to compute these parameters internally if possible, such that the method's performance would be independent of the problem. This is necessary to provide a method which can be used simply and reliably for many engineering applications. Effects of scaling, variable bounds, and perturbed constraints were not investigated but will be discussed briefly for completeness.

The choice of the initial penalty parameter, c^0 , can have significant effect on the efficiency of the multiplier method. No universally acceptable method has been found to select an initial penalty parameter, c^0 . Two methods have been found to provide good initial estimates and are recommended here: (1) choose the initial penalty parameter, c^0 , of the same order of magnitude as the initial objective function, or (2) choose the initial penalty parameter such that $|\nabla f| = c |\nabla g_{\max}|$. The second method requires the gradient of each constraint which is not directly available in the current program. Both methods have provided acceptable results for the examples tested. They cannot guarantee a reasonable convergence or a reliable answer in all cases.

Bertsekas [22] recommends as increase factor, γ not much greater than 1. A $\gamma=2$ worked well in all the cases tested. Reasonable success was achieved with $\gamma=5$, but caution should be used since instability and ill-conditioning occurred when c^0 was chosen too large. Bertsekas' recommendation of $\gamma>1$ to a maximum of $\gamma=2$ appears to be adequate in most cases.

The maximum penalty parameter, c_{\max} is not as critical to problem stability as the initial c^0 . It was noted that ill-conditioning does occur for c_{\max} too large. A good guideline is to choose c_{\max} of order five to six times that of c^0 . If ill-conditioning occurs, it will be necessary to lower c_{\max} , c^0 or both.

Choosing an initial set of multipliers, $\bar{\lambda}^0$ other than zero was not investigated. The convenience of identifying active and violated constraints without additional computations when $\lambda_i^0=0$ tends to overshadow any computational advantage which may be attained by choosing a non-zero initial multiplier.

Normalization of the constraints is necessary in the function subprogram to avoid breakdown of the multiplier method. Scaling may be required for extremely ill-conditioning problems. Various scaling methods exist, all of which provide some means making the variables, objective, and constraints (or their gradients) equal to or near the same order of magnitude.

At present, the algorithm treats all side constraints or bounds as additional problem constraints. Treating side constraints directly can greatly improve the computational efficiency, as well as simplifying problem programming. Ragsdell [23] has shown success in treating variable bounds (side constraints) directly but does not offer any theoretical justification. Further research on direct treatment of side constraints is required.

The multiplier method often converges computationally to a solution that is slightly infeasible. This may not be desirable from the practical point of view. Imai [2] considered the following perturbed problem:

$$\text{Minimize } f(\bar{X}, \bar{\lambda})$$

$$\text{such that } g_i(\bar{X}, \bar{\lambda}) \leq -\epsilon \quad (\epsilon > 0, i=1, \dots, m)$$

where ϵ is some small positive number. The constraints are pushed slightly into the feasible region, so the function will terminate in the feasible region.

The augmented Lagrangian, L becomes

$$L = f + c \sum_{i=1}^m \left[\lambda_i \psi_i + \frac{c}{2} \psi_i^2 \right] \quad (3.2)$$

$$\text{where } \psi_i = \max \left[g_i + \epsilon, -\frac{\lambda_i}{c} \right]$$

Setting $\epsilon=0$, Equation 3.2 reduces to the original Lagrange function for the general inequality constrained problem. Solutions obtained with $\epsilon=0$ are only slightly infeasible, i.e., $g_i \approx +10^{-3}$; therefore, accuracy of the original Lagrange function was acceptable even though slightly infeasible.

Finally, a termination criteria was chosen such that the relative of absolute change in the objective function was less than 0.001 for three consecutive iterations, or if a preassigned number of iterations was exceeded. This was chosen since convergence of the objective function is of more interest than convergence of the multipliers, $\bar{\lambda}$.

IV. COMPUTATIONAL EXPERIENCE

The multiplier method was used to solve five test cases. Three mathematical examples were used to demonstrate the method's computational performance as compared to other methods. Two engineering examples were then solved to show the application of the multiplier method in engineering design optimization. Results of the multiplier method computations are given in Table I. Table II shows a comparison of the multiplier method to other optimization methods for each case.

A. CASE 1: THE CONSTRAINED ROSEN-SUZUKI FUNCTION [5]

$$\text{Minimize } f(x) = x_1^2 - 5x_1 + x_2^2 - 5x_2 + 2x_3^2 - 21x_3 + x_4^2 + 7x_4 + 50$$

$$\text{such that } g_1(x) = x_1^2 + x_1 + x_2^2 - x_2 + x_3^2 + x_3 + x_4^2 - x_4 - 8 = 0$$

$$g_2(x) = x_1^2 - x_1 + 2x_2^2 + x_3^2 + 2x_4^2 - x_4 - 10 \leq 0$$

$$g_3(x) = 2x_1^2 + 2x_1 + x_2^2 - x_2 + x_3^2 - x_4 - 5 = 0$$

This problem was solved for two individual cases. Case 1A solves the problem with constraints $g_1(x)$ and $g_3(x)$ as equality constraints. Case 1B treats these two constraints as inequalities. The solution to each case is given in Table I. This problem demonstrates the method's ability to solve equality constrained problems directly with greater accuracy and efficiency. As seen in Table II, the method performs

no better than the exterior penalty method or CONMIN for the inequality constrained problem. However, when the equality constraints are treated directly, the multiplier method shows significant improvement. It should be noted that CONMIN uses a polynomial interpolation one-dimensional search, which is significantly more efficient than the Golden Section search in many cases. Using the polynomial search should make the multiplier method more efficient, if reliability can be preserved. The reliability of the Golden Section search made it the more desirable choice in the development of the method.

B. CASE 2. A SIMPLE QUADRATIC FUNCTION [23]

$$\text{Minimize } f(x) = 4x_1 - x_2^2 - 12$$

$$\text{such that } g_1(x) = 25 - x_1^2 - x_2^2 = 0$$

$$g_2(x) = -10x_1 + x_1^2 - 10x_2 + x_2^2 + 34 \leq 0$$

$$g_3(x) = -x_1 \leq 0$$

$$g_4(x) = -x_2 \leq 0$$

This case was chosen as a comparison to the Sequential Unconstrained Minimization Technique, SUMT [23], using an interior penalty function method. The problem solution is given in Table I. From Table II, it can be seen that the multiplier method is comparatively accurate to the SUMT method.

C. CASE 3: THE PAVIANI FUNCTION [23]

$$\text{Minimize } f(x) = 1000 - x_1^2 - 2x_2^2 - x_3^2 - x_1x_2 - x_1x_3$$

$$\text{such that } g_1(x) = x_1^2 + x_2^2 + x_3^2 - 25 = 0$$

$$g_2(x) = 8x_1 + 4x_2 + 7x_3 - 56 = 0$$

$$g_3(x) = -x_1 \leq 0$$

$$g_4(x) = -x_2 \leq 0$$

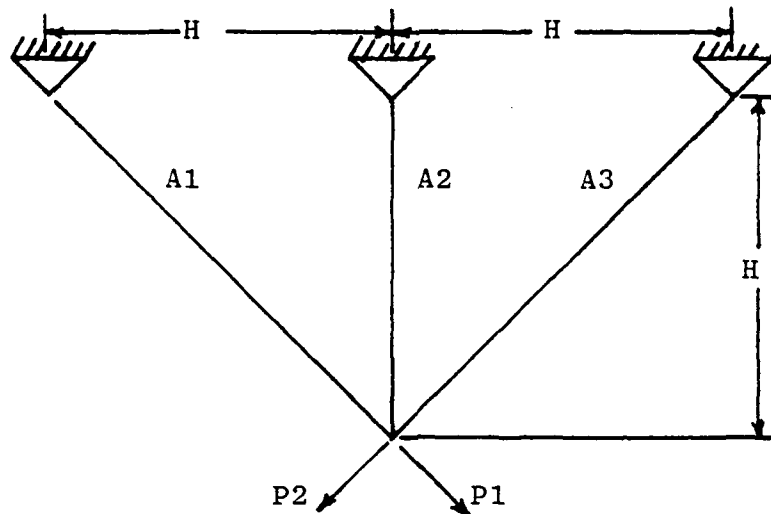
$$g_5(x) = -x_3 \leq 0$$

In this case, it is interesting to note that even though the accuracy and efficiency is comparable, the penalty method required a larger penalty parameter, c than the multiplier method. This is a common point and shows the multiplier method's ability to avoid ill-conditioning by efficiently and accurately obtaining a solution with a finite c .

D. CASE 4: THE THREE BAR TRUSS PROBLEM

As a simple structural design problem, the three bar truss in Fig. 8 was considered. The problem was to determine the areas A_1 , A_2 , A_3 to minimize the structure weight, W . The design was subject to constraints $-15000 < \sigma_{ij} < 20000$ psi where σ_{ij} is the stress in the truss member, i under load condition, j . An additional geometric constraint of $A_1 = A_3$ was imposed to maintain symmetry.

THE THREE BAR TRUSS PROBLEM



MATERIAL: $\rho = 0.1 \text{ lb/in}^3$

$E = 10^6 \text{ psi}$

GEOMETRY: $H = 10 \text{ inches}$

LOADS: $P_1 = P_2 = 20,000 \text{ lbs.}$

STRESS LIMITS: $-15,000 \leq \sigma_{ij} \leq 20,000 \text{ psi}$

INITIAL AREAS: $A1 = A2 = A3 = 1.0 \text{ sq.in.}$

FIGURE 8.

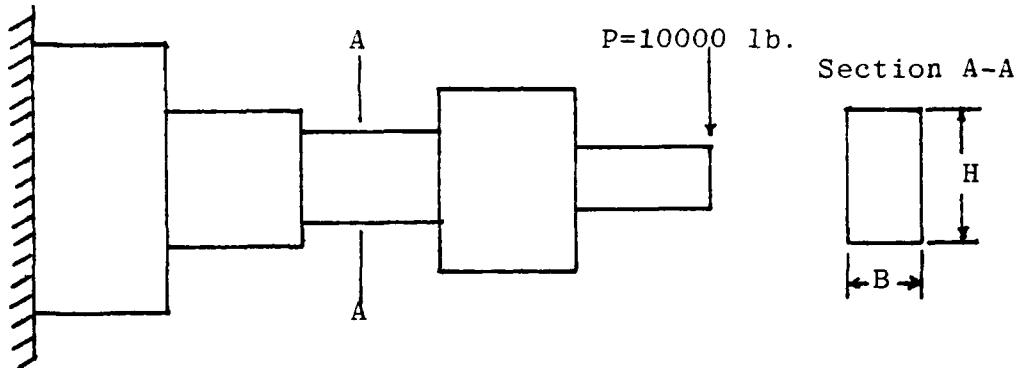
E. CASE 5: CANTILEVER BEAM PROBLEM

A cantilever beam with base, B , and height, H , variable over the length, L , was optimized to find the minimum volume. The beam, as seen in Fig. 9, was divided into five sections. The following constraints were imposed: (1) bending stress, σ in each section was not to exceed ± 20000 psi. (2) deflection, δ under the load was not to exceed ± 2.0 inches, and (3) height to beam ratio, $(H/B) < 30$. Additional side constraints were also imposed on the base and height. Separating the beam into five sections expanded the problem to one of ten design variables and 37 inequality constraints. The problem was solved by the multiplier method with $c=1000$ and $\gamma=1$. The solution is given in Table I, with a comparison to the exterior penalty method and CONMIN in Table II. It should be noted that the exterior penalty method did not obtain a solution due to numerical ill-conditioning and stability problems.

F. SUMMARY

For the cases involving equality constraints, the advantages of the multiplier method over penalty methods and CONMIN has been shown. The multiplier method has proven to be a suitable alternative to the other methods tested for both equality and inequality constrained problems. While the method may have worked well for the cases tested, it cannot be guaranteed to be the most efficient, if even the most accurate, in all cases. The dependence of the multiplier

THE CANTILEVER BEAM PROBLEM



The beam is broken into five segments of equal length and each segment is governed by two design variables for a total of ten design variables.

Design requirements:

MATERIAL: $E = 30 \times 10^6 \text{ psi}$

GEOMETRY: Total length, $L = 200 \text{ inches}$

$$1 \leq H_i \leq 30 \text{ inches} \quad i = 1, 5$$

$$0.5 \leq B_i \leq 5 \text{ inches} \quad i = 1, 5$$

$$H_i/B_i \leq 30 \quad i = 1, 5$$

STRESS LIMITS: (at the left end of each segment)

$$\sigma_i = \frac{Mc}{I} \leq 20000 \text{ psi} \quad i = 1, 5$$

FIGURE 9

method on internal programming parameters, c and λ , may even make the method less attractive in some cases to other optimization methods.

TABLE I
MULTIPLIER METHOD RESULTS

Case 1A: The Equality Constrained Rosen-Suzuki Function

Initial Obj = 31.000

$$\bar{X}^T = (1.0, 1.0, 1.0, 1.0)$$

$$\bar{G}^T = (-4.0, -6.0, -1.0)$$

Final Obj = 6.0075

$$\bar{X}^T = (0.01607, 1.0285, 1.9799, -1.018)$$

$$\bar{G}^T = (-0.00515, -0.8896, 0.004387)$$

Theoretical Optimum

Obj = 6.0000

$$\bar{X}^T = (0.0, 1.0, 2.0, -1.0)$$

$$\bar{G}^T = (0.0, -1.0, 0.0)$$

Case 1B: The Inequality Constrained Rosen-Suzuki Function

Initial Obj = 31.000

$$\bar{X}^T = (1.0, 1.0, 1.0, 1.0)$$

$$\bar{G}^T = (-4.0, -0.0, -1.0)$$

Final Obj = 6.1178

$$\bar{X}^T = (0.03206, 1.0334, 1.9509, -1.054)$$

$$\bar{G}^T = (-0.1166, -0.8921, -0.1237)$$

Theoretical Optimum = Same as above

TABLE I (contd)

Case 2: A Simple Quadratic Function

Initial Obj = -9.0

$$\bar{X}^T = (1.0, 1.0)$$

$$\bar{G}^T = (23.0, 16.0, -1.0, -1.0)$$

Final Obj = -31.989

$$\bar{X}^T = (1.0019, 4.8986)$$

$$\bar{G}^T = (-0.0017, -0.192, -1.026, -4.894)$$

Theoretical Optimum

Obj = -32.000

$$\bar{X}^T = (1.0, 4.8990)$$

$$\bar{G}^T = (0.0, 0.0102, -1.0, -4.899)$$

Case 3: The Paviani Function

Initial Obj = 976.0

$$\bar{X}^T = (210, 2.0, 2.0)$$

$$\bar{G}^T = (-13.0, 2.0, -2.0, -2.0, -2.0)$$

Final Obj = 961.79

$$\bar{X}^T = (3.289, 0.2403, 3.7599)$$

$$\bar{G}^T = (0.0035, -0.0048, -3.286, -0.241, -3.761)$$

Theoretical Optimum

Obj = 961.715

$$\bar{X}^T = (3.512, 0.217, 3.552)$$

$$\bar{G}^T = (0.0, 0.0, -3.512, -0.217, -3.552)$$

TABLE I (contd)

Case 4: The Three Bar Truss

Initial Obj = 3.8284 lbs.

$$\bar{X}^T = (1.0, 1.0, 1.0) \text{ in.}$$

$$\sigma_{11} = 14142.2 \text{ psi}$$

$$\sigma_{21} = 8284.2$$

$$\sigma_{31} = -5858.0$$

$$\sigma_{12} = -5858.0$$

$$\sigma_{22} = 8284.2$$

$$\sigma_{32} = 14142.2$$

Final Obj = 2.639

$$\bar{X}^T = (0.7885, 0.4086, 0.7883)$$

$$\sigma_{11} = 20001.$$

$$\sigma_{21} = 14636.$$

$$\sigma_{31} = -5364.8$$

$$\sigma_{21} = -5364.8$$

$$\sigma_{22} = 14636.$$

$$\sigma_{32} = 20005.$$

Theoretical Optimum

Obj = 2.632

$$\bar{X}^T = (0.781, 0.424, 0.781)$$

σ_{11} and σ_{32} are active constraints.

TABLE I (contd)

Case 5: Cantilever Beam

Initial Obj = 9000 cu. ft.

$$\bar{H}^T = (15.0, 15.0, 15.0, 15.0, 15.0)$$

$$\bar{B}^T = (3.0, 3.0, 3.0, 3.0, 3.0)$$

$$\bar{\sigma}^T = (17778, 14222, 10667, 7111.2, 3555.6)$$

$$(H/B)^T = (5.0, 5.0, 5.0, 5.0, 5.0)$$

$$\delta = 1.0535$$

Final Obj = 3206.2 cu. ft.

$$\bar{H}^T = (26.05, 24.62, 22.70, 19.83, 15.35)$$

$$\bar{B}^T = (0.886, 0.796, 0.733, 0.646, 0.506)$$

$$\bar{\sigma}^T = (19961, 19904, 19071, 18914, 20111)$$

$$(H/B)^T = (29.42, 30.92, 30.97, 30.72, 30.32)$$

$$\delta = 0.9528$$

TABLE II
COMPARISON OF VARIOUS OPTIMIZATION METHODS

	1A	1B	2	3	4	5
Number of Design Variables	4	4	2	3	4	5
Number of Equality Constraints	2	0	1	2	1	10
Number of Inequality Constraints	1	3	3	3	12	36
AUGMENTED LAGRANGE MULTIPLIER METHOD						
Number of Iterations	6	9	5	4	6	25
Number of Function Evaluations	304	347	198	120	185	3390
Objective Function Value	6.0075	6.1178	-31.989	961.79	2.6385	3206.2
EXTERIOR PENALTY FUNCTION METHOD						
Number of Iterations	7	7	5	4	10	(2)
Number of Function Evaluations	414	319	213	120	310	
Objective Function Value	6.0065	6.0165	-31.964	961.79	2.6390	
CONSTRAINED FUNCTION MINIMIZATION (CONMIN)						
Number of Iterations	(1)	11		(1)	8	20
Number of Function Evaluations		68			36	245
Objective Function Value		6.0183			2.7204	3227.5
INTERIOR PENALTY FUNCTION METHOD (SUMT)						
Number of Iterations			8			
Number of Function Evaluations			68			
Objective Function Value			-31.990	961.72		

(1) Cannot be solved by this method.

(2) Breaks down due to ill-condition.

V. CONCLUSIONS

The multiplier method has been shown to be an accurate and efficient method for solving problems in engineering design optimization. In the particular cases tested, it showed comparable or improved performance to other optimization methods. In using a finite penalty parameter, c^0 , the numerical ill-conditioning of penalty methods is avoided in most cases. The convergence of the multiplier method is at least linear while penalty methods converge asymptotically. Exact solutions are also attainable by the multiplier method. The ability to handle equality constraints directly makes it an attractive alternative to CONMIN for the equality constrained problem.

The multiplier method has other advantages which make it attractive. It can be used as an interior or exterior optimization method, i.e., the optimum can be approached from the feasible or infeasible region. Any reasonable starting point can be used. The dynamic selection of active constraints when $\bar{\lambda}^0=0$ is also a feature of the multiplier method.

Research of multiplier methods is far from complete. Its use in the various disciplines of engineering design are endless. A multiplier algorithm needs to be developed which is independent of internal computational parameters. The algorithm needs to be streamlined for easy application by the

practicing engineer. The specific applications for which the multiplier method is most attractive need to be identified.

The Augmented Lagrange Multiplier method is an extremely useful program in computer-aided engineering design. Its applications today are few, but its possibilities are endless and invaluable.

LIST OF REFERENCES

1. Fox, R.L., Optimization Methods for Engineering Design, Addison-Wesley, 1971.
2. Imai, K., Configuration Optimization of Trusses by the Multiplier Method, Ph.D. Thesis, University of California at Los Angeles, June 1978.
3. Fax, D.H., and Mills, R.R., Jr., Transactions of ASME, V. 79, pp. 653-661, 1957.
4. Hedderich, C.P., Heat Exchanger Optimization, M.S. Thesis, Naval Postgraduate School, Monterey, California, September 1980.
5. Vanderplaats, G.N., NASA Ames Research Center Technical Memorandum NASA TM X-62,282, CONMIN - A FORTRAN Program for Constrained Function Minimization User's Manual, August 1973.
6. Hestenes, M.R., Optimization Theory, Wiley, 1975.
7. Hestenes, M.R., "Multiplier and Gradient Methods," Journal of Optimization Theory and Applications (JOTA), V. 4, No. 5, pp. 303-320, 1969.
8. Fletcher, R., Optimization, pp. 283-298, Academic Press, 1969.
9. Miele, A., and others, "Use of the Augmented Penalty Function in Mathematical Programming Problems, Part I and II, JOTA, V. 8, No. 2, pp. 115-153, 1971.
10. Miele, A., and others, "On the Method of Multipliers for Mathematical Programming Problems," JOTA, V. 10, No. 1, pp. 1-33, 1972.
11. Tripathi, S.S., and Navendra, K.S., "Constrained Optimization Problems Using Multiplier Methods," JOTA, V. 9, No. 1, pp. 59-70, 1972.
12. Rockafellar, R.T., "Augmented Lagrange Multiplier Functions and Duality in Non-Convex Programming," SIAM J. of Control, V. 12, No. 2, pp. 268-285, May 1974.

13. Rockafellar, R.T., "The Multiplier Method of Hestenes and Powell Applied to Convex Programming," JOTA, V. 12, No. 6, pp. 555-562, 1973.
14. Bertsekas, D.P., "Combined Primal-Dual and Penalty Methods for Constrained Minimization," SIAM J. of Control, V. 13, No. 3, pp. 521-544, May 1975.
15. Bertsekas, D.P., "On Penalty and Multiplier Methods for Constrained Minimization," SIAM J. of Control and Optimization, V. 14, No. 2, pp. 216-235, February 1976.
16. Luenberger, D.G., Introduction to Non-Linear Programming, Addison-Wesley, 1973.
17. Schmit, L.A. and Fleury, C., "An Improved Analysis/Synthesis Capability Based on Dual Methods-Access 3," AIAA 20th Structures, Structural Dynamics, and Materials Conference, pp. 23-50, April 1979.
18. Pierre, D.A., and Lowe, M.J., Mathematical Programming via Augmented Lagrangians, Addison-Wesley, 1975.
19. Haftka, R.T. and Starnes, J.H., "Application of a Quadratic Extended Interior Penalty Function for Structural Optimization," AIAA Journal, V. 14, No. 6, pp. 718-724, June 1976.
20. Buys, J.D., Dual Algorithms for Constrained Optimization, Ph.D. Thesis, Rijksuniversiteit de Leiden, 1972.
21. Rupp, R.D., "Approximation of the Classical Isoperimetric Problem," JOTA, V. 9, pp. 251-264, 1972.
22. Bertsekas, D.P., "Multiplier Methods: A Survey," Automatica, V. 12, pp. 133-145, 1976.
23. Ragsdell, K.M., and Root, R.R., "Computational Enhancements to the Method of Multipliers," TRANS. ASME, J. of Mechanical Design, pp. 517-523, July 1980.
24. Himmelblau, D.M., Applied Nonlinear Programming, McGraw-Hill, 1972.

APPENDIX A

ALGORITHM FOR GOLDEN SECTION ONE-DIMENSIONAL SEARCH

The algorithm for the one-dimensional search subprogram is outlined in this appendix. The one-dimensional search is first performed by the Golden Section method. A cubic approximation is then performed on the last four values found by Golden Section method. This approach proved to be very effective in obtaining an accurate solution. The one-dimensional search is performed as follows:

Step 1. Specify the initial interval x_l and x_u , where x denotes the scalar, α , in Equation 3.1.

Evaluate the functions $y_l = f(x_l)$, $y_u = f(x_u)$.

Step 2. Specify the absolute convergence tolerance, Δx . Calculate the relative convergence tolerance, ϵ , and the number of function evaluations, N , from the equations.

$$\epsilon = \Delta x / (x_u - x_l)$$

$$N = -2.078 \ln(\epsilon) + 3$$

The value N is calculated in floating point arithmetic and rounded off to the next higher integer. If $N < 4$, set $N = 4$.

Step 3. Calculate x_1 and x_2 by the equations

$$x_1 = (1 - \tau) x_l + \tau x_u$$

$$x_2 = \tau x_l + (1 - \tau) x_u$$

where $\tau = (3 - \sqrt{5})/2 = 0.38197$

Evaluate $y_1 = f(x_1)$ and $y_2 = f(x_2)$.

Step 4: Set counter, $K=4$. This is because four functions have already been evaluated. If $N=4$, go to Step 9.

Step 5: If y_2 is greater than y_1 , go to Step 7.

Step 6: y_1 is greater than or equal to y_2 . x_1 is a new lower bound. Set:

$$\begin{aligned}x_l &= x_1 \\y_l &= y_1 \\x_1 &= x_2 \\y_1 &= y_2 \\x_2 &= \tau x_l + (1-\tau)x_u \\y_2 &= f(x_2) \\K &= K + 1\end{aligned}$$

Go to Step 8.

Step 7: y_2 is greater than y_1 . x_2 is a new upper bound. Set:

$$\begin{aligned}x_u &= x_2 \\y_u &= y_2 \\x_2 &= x_1 \\y_2 &= y_1 \\x_1 &= (1-\tau)x_l + \tau x_u \\y_1 &= f(x_1) \\K &= K + 1\end{aligned}$$

Step 8: Check convergence. If $K > N$, go to Step 9; else go to Step 5.

Step 9: Do cubic approximation with values x_l, x_1, x_2, x_u .

$$y_c = f(x_l, x_1, x_2, x_u)$$

Step 10: Pick best of y_l, y_1, y_2, y_c .

$$Y = \min (y_l, y_1, y_2, y_c)$$

X = corresponding x_l, x_1, x_2 or x_c

Y is the optimum function value at location X . Stop.

APPENDIX B

THIS IS A SAMPLE OUTPUT OF THE THREE BAR TRUSS PROBLEM AS SOLVED BY THE AUGMENTED LAGRANGE MULTIPLIER METHOD. AN INITIAL PENALTY PARAMETER C=10 AND GAMMA=2 WAS USED. THE PROBLEM SUBPROGRAM IS LISTED FIRST, FOLLOWED BY THE RESULTS.

C 3-BAR TRUSS PROBLEM

SUBROUTINE FUNC (ICALC,N,X,FX)
COMMON /LAGR/ NCON,G(40),ISC(40),XLAM(40),C
COMMON /CTRL/ CNTR,NDRV,NFUNC,IPRINT
DIMENSION X(3)
NCON=13

C N=3
IF (ICALC.GT.1) GO TO 10
INPUT OR INITIALIZATION

X(1)=1.
X(2)=1.
X(3)=1.
E=1.E7
RHO=.1
H=10.
P=20000.

WRITE (6,30) X(1),X(2),X(3),E,RHO,H,P

10 C RETURN
EXECUTION

NFUNC=NFUNC+1
W=RHO*H*(SQRT(2.)*(X(1)+X(3))+X(2))
DEN=X(1)*X(2)+X(2)*X(3)+SQRT(2.)*X(1)*X(3)
SIG11=P/X(1)-X(2)*X(3)*P/(X(1)*DEN)
SIG21=SQRT(2.)*X(3)*P/DEN
SIG31=-X(2)*P/DEN
SIG12=-X(2)*P/DEN
SIG22=SQRT(2.)*X(1)*P/DEN
SIG32=P/X(3)-X(1)*X(2)*P/(X(3)*DEN)

FX=W
G(1)=SIG11/20000.-1.
G(2)=SIG21/20000.-1.
G(3)=SIG31/20000.-1.
G(4)=SIG12/20000.-1.
G(5)=SIG22/20000.-1.
G(6)=SIG32/20000.-1.
G(7)=-SIG11/15000.-1.
G(8)=-SIG21/15000.-1.
G(9)=-SIG31/15000.-1.

APP000040
APP000050
APP000060
APP000070
APP000080
APP000090
APP000100
APP000110
APP000120
APP000130
APP000140
APP000150
APP000160
APP000170
APP000180
APP000190
APP000200
APP000210
APP000220
APP000230
APP000240
APP000250
APP000260
APP000270
APP000280
APP000290
APP000300
APP000310
APP000320
APP000330
APP000340
APP000350
APP000360
APP000370
APP000380
APP000390
APP000400
APP000410
APP000420
APP000430
APP000440
APP000450
APP000460
APP000470
APP000480
APP000490
APP000500
APP000510


```

UNCONSTRAINED MINIMIZATION RESULTS
FX 0.24984E+01
X-VECTOR
0.71042E+00 0.37348E+00 0.71042E+00
NUMBER OF ITERATIONS= 7
TERMINATION CRITERION
ABS(1-FX/FY).LT..001 FOR 2 CONSECUTIVE ITERATIONS
COUNTER NUMBER 2 CMAX = 0.10000E+07 GAMMA = 0.20000E+01
C = 0.20000E+02
X-VECTOR
0.71042E+00 0.37348E+00 0.71042E+00
OBJ = 0.24984E+01
CONSTRAINT VALUES
0.10749E+00 -0.19264E+00 -0.13001E+01 -0.19264E+00
0.10749E+00 -0.24766E+01 -0.20765E+01 -0.59984E+00
-0.20765E+01 -0.24766E+01 0.0
LAMBDA VECTOR
0.10749E+01 0.0 0.0 0.0
0.10749E+01 0.0 0.0 0.0
0.0 0.0 0.0 0.0
UNCONSTRAINED MINIMIZATION RESULTS
FX 0.26364E+01
X-VECTOR
0.78009E+00 0.40891E+00 0.78009E+00
NUMBER OF ITERATIONS= 3
TERMINATION CRITERION
ABS(1-FX/FY).LT..001 FOR 2 CONSECUTIVE ITERATIONS
ABS(FX-FY).LT..001 FOR 2 CONSECUTIVE ITERATIONS
COUNTER NUMBER 3 CMAX = 0.10000E+07 GAMMA = 0.20000E+01
C = 0.40000E+02
X-VECTOR
0.78009E+00 0.40891E+00 0.78009E+00
OBJ = 0.26364E+01
CONSTRAINT VALUES
0.90408E-02 -0.26382E+00 -0.12729E+01 -0.26382E+00
0.90408E-02 -0.23454E+01 -0.19816E+01 -0.63618E+00
-0.19816E+01 -0.23454E+01 -0.59605E-07
LAMBDA VECTOR
0.12557E+01 0.0 0.0 0.0

```

```

APP01003
APP01010
APP01020
APP01030
APP01040
APP01050
APP01060
APP01070
APP01080
APP01090
APP01100
APP01110
APP01120
APP01130
APP01140
APP01150
APP01160
APP01170
APP01180
APP01190
APP01200
APP01210
APP01220
APP01230
APP01240
APP01250
APP01260
APP01270
APP01280
APP01290
APP01300
APP01310
APP01320
APP01330
APP01340
APP01350
APP01360
APP01370
APP01380
APP01390
APP01400
APP01410
APP01420
APP01430
APP01440
APP01450
APP01460
APP01470

```

```

0 0.12557E+01 0.0 0.0 0.0 0.0 0.0
0.0 -0.11921E-05 0.0 0.0
-----
UNCONSTRAINED MINIMIZATION RESULTS
FX 0.26391E+01
X-VECTOR
0.78191E+00 0.42038E+00 0.78179E+00
NUMBER OF ITERATIONS= 2
TERMINATION CRITERION
ABS(1-FX/FY).LT..001 FOR 2 CONSECUTIVE ITERATIONS
-----
0 0 COUNTER NUMBER 4 CMAX = 0.10000E+07 GAMMA = 0.20000E+01
0 0 C = 0.80000E+02
0 0 X-VECTOR
0.78191E+00 0.42038E+00 0.78179E+00
0 0 OBJ = 0.26391E+01
0 0 CONSTRAINT VALUES
0.27313E-02 -0.27350E+00 -0.12762E+01 -0.27339E+00
0.28429E-02 -0.23370E+01 -0.19687E+01 -0.63169E+00
-0.19688E+01 -0.23371E+01 0.12040E-03
0 0 LAMBDA VECTOR
0.13649E+01 0.0 0.0 0.0
0.13694E+01 0.0 0.0 0.0
0.0 0.0 0.0 0.0
-----
UNCONSTRAINED MINIMIZATION RESULTS
FX 0.26391E+01
X-VECTOR
0.78656E+00 0.41182E+00 0.78724E+00
NUMBER OF ITERATIONS= 2
TERMINATION CRITERION
ABS(1-FX/FY).LT..001 FOR 2 CONSECUTIVE ITERATIONS
ABS(FX-FY).LT..001 FOR 2 CONSECUTIVE ITERATIONS
-----
0 0 COUNTER NUMBER 5 CMAX = 0.10000E+07 GAMMA = 0.20000E+01
0 0 C = 0.16000E+03
0 0 X-VECTOR
0.78656E+00 0.41182E+00 0.78724E+00
0 0 OBJ = 0.26391E+01
0 0 CONSTRAINT VALUES
0.86594E-03 -0.26939E+00 -0.12703E+01 -0.27002E+00
0.23937E-03 -0.23345E+01 -0.19741E+01 -0.63966E+00

```

```

-0.19733E+01 -0.23337E+01 -0.67466E-03
0 LAMBDA VECTOR
0.14342E+01 0.0 0.0 0.0
0.13885E+01 0.0 0.0 0.0
0.0 -0.49158E-01 0.0 0.0
0 UNCONSTRAINED MINIMIZATION RESULTS
FX 0.26390E+01
X-VECTOR
0.78835E+00 0.40911E+00 0.78783E+00
NUMBER OF ITERATIONS= 4
TERMINATION CRITERION
ABS(1-FX/FY).LT.0.001 FOR 2 CONSECUTIVE ITERATIONS
ABS(FX-FY).LT.0.001 FOR 2 CONSECUTIVE ITERATIONS
0 COUNTER NUMBER 6 CMAX = 0.10000E+07 GAMMA = 0.20000E+01
0 C = 0.32000E+03
0 X-VECTOR
0.78835E+00 0.40911E+00 0.78783E+00
0 OBJ = 0.26390E+01
0 CONSTRAINT VALUES
0.61035E-04 -0.26853E+00 -0.12686E+01 -0.26805E+00
0.54264E-03 -0.23334E+01 -0.19753E+01 -0.64188E+00
-0.19759E+01 -0.23341E+01 0.51999E-03
0 LAMBDA VECTOR
0.14440E+01 0.0 0.0 0.0
0.14754E+01 0.0 0.0 0.0
0.0 0.34040E-01 0.0 0.0
0 UNCONSTRAINED MINIMIZATION RESULTS
FX 0.26390E+01
X-VECTOR
0.78873E+00 0.40778E+00 0.78860E+00
NUMBER OF ITERATIONS= 4
TERMINATION CRITERION
ABS(1-FX/FY).LT.0.001 FOR 2 CONSECUTIVE ITERATIONS
ABS(FX-FY).LT.0.001 FOR 2 CONSECUTIVE ITERATIONS
0 IFINAL OPTIMIZATION INFORMATION
FX = 0.26390E+01

```

APP02440
 APP02450
 APP02460
 APP02470
 APP02480
 APP02490
 APP02500
 APP02510
 APP02520
 APP02530
 APP02540
 APP02550
 APP02560
 APP02570
 APP02580
 APP02590
 APP02600
 APP02610
 APP02620
 APP02630
 APP02640
 APP02650

X-VECTOR 0.78873E+00 0.40778E+00 0.78860E+00
 NUMBER OF ITERATIONS= 6
 FUNCTION EVALUATED 185 TIMES
 TERMINATION CRITERION
 ABS(FX-FY).LT..001 FOR 3 CONSECUTIVE ITERATIONS
 ABS(1-FX/FY).LT..001 FOR 3 CONSECUTIVE ITERATIONS
 X(1)= 0.78873E+00
 X(2)= 0.40778E+00
 X(3)= 0.78860E+00
 SIG11= 0.20002E+05
 SIG21= 0.14647E+05
 SIG31= -0.53556E+04
 SIG12= -0.53556E+04
 SIG22= 0.14649E+05
 SIG32= 0.20005E+05
 WEIGHT= 0.26385E+01

0
 0
 1

APPENDIX C

THIS IS A LISTING OF THE AUGMENTED LAGRANGE MULTIPLIER PROGRAM AND ITS ASSOCIATED SUBPROGRAMS. THE PROGRAM REQUIRES AN ADDITIONAL USER SUPPLIED SUBPROGRAM TO INPUT THE DESIRED PROBLEM. THE PROGRAM MUST BE OF THE FOLLOWING FORM:

```

C 10
C
SUBROUTINE FCN (ICALC,N,X,FX)
COMMON/LAGR/ NCON,G(40),ISC(40),XLAM(40),C
COMMON/CTRL/ CNTR,NDRV,NFUNC,IPRINT
IF (ICALC.GT.1)GO TO 10OBJECTIVE FUNCTION. THE SECOND CARD IS,
INPUT OR INITIALIZATION OF DATA
CONTINUE
EXECUTION
(USER SUPPLIED OBJECTIVE AND NORMALIZED CONSTRAINT FUNCTIONS)
(DEFINE CONSTRAINT TYPE SETTING ISC(I)= 1 FOR EQUALITY CONSTRAINT
OF ISC(I)= 3 FOR INEQUALITY CONSTRAINT, WHERE I IS THE CONSTRAINT
NUMBER)
NFUNC= NFUNC+1
IF (ICALC.LT.3) RETURN
OUTPUT
RETURN
END
C

```

THE FOLLOWING IS THE AUGMENTED LAGRANGE MULTIPLIER PROGRAM.

```

C
C MAIN PROGRAM FOR EXECUTION OF LAGRANGE MULTIPLIER METHOD
C
DIMENSION X(10),Y(10),S(10),H(10,10),GX(10),GY(10),DELG(10),DELX(
10)
COMMON /LAGR/ NCON,G(40),ISC(40),XLAM(40),C
EXTERNAL LAGR1
NDV=10
NH=10
CALL LGRANG (NDV,X,FX,Y,S,H,NH,GX,GY,DELG,DELX)
STOP
END
SUBROUTINE LGRANG (NDV,X,FX,Y,S,H,NH,GX,GY,DELG,DELX)
*****
FUNCTION MINIMIZATION BY AUGMENTED LAGRANGE MULTIPLIER METHOD
*****
BY L. E. MADSEN
NAVAL POSTGRADUATE SCHOOL, MONTEREY, CALIF.
*****
PRINT CONTROL
0- INPUT/OUTPUT ONLY
C

```

APP000040
APP000050
APP000060
APP000070
APP000080
APP000090
APP000100
APP000110
APP000120
APP000130
APP000140
APP000150
APP000160
APP000170
APP000180
APP000190
APP000200
APP000210
APP000220
APP000230
APP000240
APP000250
APP000260
APP000270
APP000280
APP000290
APP000300
APP000310
APP000320
APP000330
APP000340
APP000350
APP000360
APP000370
APP000380
APP000390
APP000400
APP000410
APP000420
APP000430
APP000440
APP000450
APP000460
APP000470
APP000480
APP000490
APP000500
APP000510


```

40 KOBJ2=KOBJ2+1
C IF (ABS(1.-FX/(FY+1.E-10)).GE..001) KOBJ2=0
C IF (KOBJ1.GE.ITRM.OR.KOBJ2.GE.ITRM) GO TO 100
D0 40 I=1,NDV
X(I)=Y(I)
CONTINUE
FX=FY

C UPDATE LAMBDA VECTOR
IF (LFLG.GE.1) GO TO 90
D0 80 I=1,NCON
II=ISC(I)
IF (II.GT.2) GO TO 50
INEQUALITY CONSTRAINT
PSI=-XLAM(I)/C
IF (G(II).GT.PSI) PSI=G(II)
XLAM(I)=XLAM(II)+C*PSI
GO TO 70
IF (II.GT.4) GO TO 60
EQUALITY CONSTRAINT
XLAM(II)=XLAM(I)+C*G(II)
GO TO 70
WRITE (6,180) II,ISC(II)
IF (ABS(XLAM(I)).LT.1.E-15) XLAM(I)=0.
IF (XLAM(I).GE.1.E15) XLAM(I)=1.E15
IF (XLAM(I).LT.-1.E15) XLAM(I)=-1.E15
CONTINUE
C=GAMMA*C
IF (C.GT.CMAX) C=CMAX
GO TO 20
WRITE (6,190) FY
WRITE (6,140) (Y(I),I=1,NDV)
WRITE (6,200) NCTR,NFUNC
IF (KOBJ3.GT.0) WRITE (6,210) ITRM
IF (KOBJ1.GT.0) WRITE (6,220) ITRM
ICALLC=3
CALL FUNC (ICALLC,NDV,Y,FY)
RETURN

C 110 FORMAT (1H0,15H COUNTER NUMBER,15)
120 FORMAT (1H0,4X,4HC = ,E12.5,5X,7HCMAX = ,E12.5,5X,8HGGAMMA = ,E12.5
1)
130 FORMAT (1H0,4X,8HX-VECTOR)
140 FORMAT (1H,5(5X,E12.5))
150 FORMAT (1H0,4X,6H0BJ = ,E12.5)
160 FORMAT (1H0,4X,17HCONSTRAINT VALUES)
170 FORMAT (1H0,4X,13HLAMBDA VECTOR)

```

```

180 FORMAT (1H0,23HERROR: ISC,GT,4: ISC(,13,3H)=,I3) APP01480
190 FORMAT (1H1,30HFINAL OPTIMIZATION INFORMATION//5X,5HFx = ,E12.5//5A APP01490
200 1X,8HX-VECTOR) APP01500
210 1ED,15,6H TIMES//5X,21HTERMINATION CRITERION) APP01510
220 1FORMAT (1H0,4X,22HNUMBER OF ITERATIONS=,I5/5X,18HFUNCTION EVALUAT APP01520
230 1ED,15,6H TIMES//5X,21HTERMINATION CRITERION) APP01530
240 1FORMAT (1H0,9X,37HMAXIMUM NUMBER OF ITERATIONS EXCEEDED) APP01540
250 1FORMAT (1H0,9X,22HABS(FX-FY).LT..001 FOR,I2,23H CONSECUTIVE ITERAT APP01550
260 1IONS) APP01560
270 1FORMAT (1H0,9X,25HABS(1.-FX/FY).LT..001 FOR,I2,23H CONSECUTIVE ITERAT APP01570
280 1RATIONS) APP01580
290 1END APP01590
300 1SUBROUTINE LAGR1 (NDV,X,OBJ) APP01600
310 1DIMENSION X(10) APP01610
320 1COMMON /LAGR/ NCON,G(40),ISC(40),XLAM(40),C APP01620
330 1COMMON /CTRL/ NCTR,NDRV,NFUNC,IPRINT APP01630
340 1ICALLC=2 APP01640
350 1CALL FUNC (ICALLC,NDV,X,OBJ) APP01650
360 1IF (IPRINT.GE.9) WRITE (6,40) OBJ APP01660
370 1IF (IPRINT.GE.9) WRITE (6,50) (G(I),I=1,NCON) APP01670
380 1DO 30 I=1,NCON APP01680
390 1II=ISC(I) APP01690
400 1IF (II.GT.2) GO TO 10 APP01700
410 1INEQUALITY CONSTRAINT APP01710
420 1PSI=-XLAM(I)/C APP01720
430 1IF (G(I).GT.PSI) PSI=G(I) APP01730
440 1OBJ=OBJ+XLAM(I)*PSI+.5*C*PSI**2 APP01740
450 1GO TO 30 APP01750
460 1IF (II.GT.4) GO TO 20 APP01760
470 1EQUALITY CONSTRAINT APP01770
480 1OBJ=OBJ+XLAM(I)*G(I)+.5*C*G(I)**2 APP01780
490 1GO TO 30 APP01790
500 1CONSTRAINT IS INDETERMINATE APP01800
510 1WRITE (6,60) ISC(I) APP01810
520 1IF (IPRINT.GE.9) WRITE (6,70) OBJ APP01820
530 1CONTINUE APP01830
540 1RETURN APP01840
550 1FORMAT (1H0,4X,4HEX=,E12.5//5X,17HCONSTRAINT VALUES) APP01850
560 1FORMAT (1H,5(3X,E12.5)) APP01860
570 1FORMAT (1H0,34HERROR: ISC UNDEFINED. ISC VALUES/5(3X,E12.5)) APP01870
580 1FORMAT (1H0,4X,5HOBJ=,E12.5) APP01880
590 1END APP01890
600 1SUBROUTINE DFPEX (N,X,Y,S,H,NH,GX,GY,DELG,DELX,FY,FCN) APP01900
610 1***** APP01910
620 1ROUTINE FOR DAVIDON-FLETCHER-POWELL METHOD FOR UNCONSTRAINED APP01920
630 1MINIMIZATION APP01930
640 1***** APP01940
650 1BY L. E. MADSEN MARCH,1981 APP01950

```


APP02440
APP02450
APP02460
APP02470
APP02480
APP02490
APP02500
APP02510
APP02520
APP02530
APP02540
APP02550
APP02560
APP02570
APP02580
APP02590
APP02600
APP02610
APP02620
APP02630
APP02640
APP02650
APP02660
APP02670
APP02680
APP02690
APP02700
APP02710
APP02720
APP02730
APP02740
APP02750
APP02760
APP02770
APP02780
APP02790
APP02800
APP02810
APP02820
APP02830
APP02840
APP02850
APP02860
APP02870
APP02880
APP02890
APP02900
APP02910

```

DO 40 I=1,N
S(I)=-GX(I)
SMAX=ABS(S(I))
IF (SMAX.GT.SCAL) SCAL=SMAX
CONTINUE
NEWH=NEWH+1
IF (NEWH.GE.ICNDIR) GO TO 10
ITER=ITER+1
IF (IPRINT.GE.2) WRITE (6,150) ITER
IF (IPRINT.GE.2) WRITE (6,160) (GX(I),I=1,N)
IF (IPRINT.GE.8) WRITE (6,180) (S(I),I=1,N)
IF (IPRINT.GE.8) WRITE (6,190)
DO 50 I=1,N
IF (IPRINT.GE.8) WRITE (6,170) (H(I,J),J=1,N)
CONTINUE
SDF=0.
DO 60 I=1,N
SDF=SDF+S(I)*GX(I)
CONTINUE
IF (IPRINT.GE.8) WRITE (6,200) SDF
IF (SDF.GE.0.) GO TO 10
SCAL=SCAL
SCAL1=0.
IF (IPRINT.GE.8) WRITE (6,210) SCAL
DO 70 I=1,N
S(I)=S(I)/SCAL
CONTINUE
FIND NEXT POINT, X
CALL GOLDS (N,X,FX,S,Y,FY,FCN)
IF (IPRINT.GE.2) WRITE (6,220) FY
IF (IPRINT.GE.2) WRITE (6,170) (Y(I),I=1,N)
IF (FX.GE.FY) GO TO 80
KOUNT=KOUNT+1
IF (KOUNT.GE.2) GO TO 80
IF (IPRINT.GE.8) WRITE (6,230)
GO TO 10
CONTINUE
CHECK CONVERGENCE
IF (ITER.LT.ITMAX) GO TO 90
KOBJ3=1
GO TO 130
KOBJ1=KOBJ1+1
IF (ABS(1.-FX/(FY-1.E-9)).GE..001) KOBJ1=0
KOBJ2=KOBJ2+1

```

40

50

60

70
C
C

80
C
C

90

```

C
100
C
110
C
120
C
130
C
140
150
160
170

IF (ABS(FX-FY).GE.0.001) KOBJ2=0
IF (KOBJ1.GE.ITRM.OR.KOBJ2.GE.ITRM) GO TO 130

CONVERGENCE CRITERIA NOT SATISFIED. FIND A NEW DIRECTION MATRIX
CALL DERIV (N,Y,DX,FY,GY,FCN)
DO 100 I=1,N
DELG(I)=GY(I)-GX(I)
DELX(I)=Y(I)-X(I)
GX(I)=GY(I)
X(I)=Y(I)
FX=FY
IF ((KOBJ1.EQ.0.OR.KOBJ2.EQ.0).AND.IFLAG.NE.1) GO TO 10

CHECK FY FOR MINIMUM IN S-DIRECTION. SKIP H UPDATE, IF NOT.
SDF2=0.
DO 110 I=1,N
SDF2=SDF2+$(I)*GY(I)
CONTINUE
IF (IPRINT.GE.8) WRITE (6,240) SDF2
IFLAG=IFLAG+1
IF (ABS(SDF2).LT.0.01.OR.IFLAG.GE.2) IFLAG=0
IF (IFLAG.EQ.0) CALL DFP01 (N,H,NH,DELX,DELG)

SET UP NEW SEARCH DIRECTION
DO 120 I=1,N
S(I)=0.
DO 120 J=1,N
S(I)=S(I)-H(I,J)*GY(J)
SMAX=ABS(S(I))
IF (SMAX.GT.SCAL1) SCAL1=SMAX
CONTINUE
GO TO 40

CONVERGENCE TEST SATISFIED
IF (IPRINT.GE.1) WRITE (6,250) FY
IF (IPRINT.GE.1) WRITE (6,260) (Y(I),I=1,N)
IF (IPRINT.GE.1) WRITE (6,270) ITER
IF (IPRINT.GE.1) WRITE (6,270) IITER
IF (IPRINT.GE.1.AND.KOBJ3.GT.0) WRITE (6,280) ITRM
IF (IPRINT.GE.1.AND.KOBJ1.GT.0) WRITE (6,290) ITRM
IF (IPRINT.GE.1.AND.KOBJ2.GT.0) WRITE (6,300) ITRM
IF (IPRINT.GE.1) WRITE (6,250)
RETURN

FORMAT (1H0,39HUNCONSTRAINED MINIMIZATION: DFP METHOD)
FORMAT (1H0,10HITERATION=,15)
FORMAT (1H0,4X,15HGRADIENT VECTOR)
FORMAT (1H,5(3X,E12.5))

```

```

APP02920
APP02930
APP02940
APP02950
APP02960
APP02970
APP02980
APP02990
APP03000
APP03010
APP03020
APP03030
APP03040
APP03050
APP03060
APP03070
APP03080
APP03090
APP03100
APP03110
APP03120
APP03130
APP03140
APP03150
APP03160
APP03170
APP03180
APP03190
APP03200
APP03210
APP03220
APP03230
APP03240
APP03250
APP03260
APP03270
APP03280
APP03290
APP03300
APP03310
APP03320
APP03330
APP03340
APP03350
APP03360
APP03370
APP03380
APP03390

```

```

180      FORMAT (1H0,4X,8HS-VECTOR)
190      FORMAT (1H0,4X,8HH-MATRIX)
200      FORMAT (1H0,4X,5HSDF,EL2.5/)
210      FORMAT (1H0,4X,6HSCAL,EL2.5/)
220      FORMAT (1H0,4X,4HFX,EL2.5//5X,8HX-VECTOR)
230      FORMAT (1H0,4X,19HFY,GT.FX,RESET H.)
240      FORMAT (1H0,4X,6HSDF2,EL2.5)
250      FORMAT (1H0,4X,68(1H-))
260      FORMAT (1H,4X,34HUNCONSTRAINED MINIMIZATION RESULTS//5X,4HFX,ELAPPO3480
270      12.5//5X,8HX-VECTOR)
280      FORMAT (1H0,4X,21HNUMBER OF ITERATIONS=,15//5X,21HTERMINATION CRITAPPO3490
290      1ATIONS)
300      FORMAT (1H,9X,37HMAXIMUM NUMBER OF ITERATIONS EXCEEDED)
      FORMAT (1H,9X,24HABS(1-FX/FY).LT..001 FOR,I2,23H CONSECUTIVE ITERAPPO3510
      IONS)
      FORMAT (1H,9X,21HABS(FX-FY).LT.001 FOR,I2,23H CONSECUTIVE ITERAPPO3520
      IONS)
      END
      SUBROUTINE DFP01 (N,H,NH,DELX,DELG)
      *****
      ROUTINE TO APPROXIMATE THE INVERSE HESSIAN MATRIX BY DAVIDON-
      FLETCHER-POWELL METHOD
      *****
      BY L. E. MADSEN
      NAVAL POSTGRADUATE SCHOOL, MONTEREY, CALIF.
      *****
      COMMON /CTRL/ NCTR,NDRV,NFUNC,IPRINT
      DIMENSION HDG(10),DGH(10),DELX(1),DELG(1),H(NH,1),A(10,10),B(10,10)APPO3630
      1,
      DO 10 I=1,N
      DO 10 J=1,N
      A(I,J)=0.
      B(I,J)=0.
      CONTINUE
      DXDG=1.E-9
      DGHDG=1.E-9
      DO 30 I=1,N
      HDG(I)=0.
      DGH(I)=0.
      DO 20 J=1,N
      HDG(I)=HDG(I)-H(I,J)*DELG(J)
      DGH(I)=DGH(I)+DGH(I)*DELG(J)
      DXDG=DXDG+DELX(I)*DELG(I)
      DGHDG=DGHDG+DGH(I)*DELG(I)
      DO 40 I=1,N
      DO 40 J=1,N
      A(I,J)=A(I,J)+DELX(I)*DELX(J)/DXDG
      B(I,J)=B(I,J)+DGH(I)*DGH(J)/DGHDG
      *****
      MARCH, 1981
      *****
      APP03400
      APP03410
      APP03420
      APP03430
      APP03440
      APP03450
      APP03460
      APP03470
      APP03480
      APP03490
      APP03500
      APP03510
      APP03520
      APP03530
      APP03540
      APP03550
      APP03560
      APP03570
      APP03580
      APP03590
      APP03600
      APP03610
      APP03620
      APP03630
      APP03640
      APP03650
      APP03660
      APP03670
      APP03680
      APP03690
      APP03700
      APP03710
      APP03720
      APP03730
      APP03740
      APP03750
      APP03760
      APP03770
      APP03780
      APP03790
      APP03800
      APP03810
      APP03820
      APP03830
      APP03840
      APP03850
      APP03860
      APP03870

```



```

C
X11=X1**2
X22=X2**2
X33=X3**2
IF (ICALC.GT.1) GO TO 10
THREE-POINT APPROXIMATION, GIVEN (X1,Y1,YP1), (X2,Y2), (X3,Y3).
A3=((X21*(Y3-Y1)/X31-X31*(Y2-Y1)/X21+X32*(Y2-Y1)/X31-X31*(Y2-Y1)/X21-(2.*X1+X2)*A3)/X1
A1=Y1-2.*A2*X1-3.*A3*X1*X1
A2=Y1-X1*(A1+X1*(A2+A3*X1))
RETURN
FOUR-POINT APPROXIMATION, GIVEN (X1,Y1), (X2,Y2), (X3,Y3),
(X4,Y4).
CONFINE
X42=X4-X2
X41=X4-X1
Q1=X33*X3*X21-X22*X2*X31+X11*X1*X32
Q2=X43*X3*X21-X22*X2*X41+X11*X1*X42
Q3=X32*X21*X31
Q4=X42*X21*X41
Q5=Y3*X21-Y2*X31+Y1*X32
Q6=Y4*X21-Y2*X41+Y1*X42
DENOM=Q2*Q3-Q1*Q4
A3=0.
IF (ABS(DENOM).GT.ZRO) A3=(Q3*Q6-Q4*Q5)/DENOM
A2=(Q5-Q1*A3)/Q3
A1=(Y2-Y1)/X21-(X22*X2-X11*X1)*A3/X21-(X1+X2)*A2
A2=Y1-X1*(A1+X1*(A2+X1*A3))
RETURN
END
SUBROUTINE ADS005(ICALC,A1,A2,A3,XMNI,XXMI)
*****
FIND MINIMUM(S) OF POLYNOMIAL TO 3RD ORDER.
*****
BY G. N. VANDERPLAATS
NAVAL POSTGRADUATE SCHOOL, MONTEREY, CALIF.
AUG. 1980
-----
INPUT
ICALC - CALCULATION CONTROL. NO MINIMUM OR
ICALC=1, FIRST ORDER POLYNOMIAL. RETURN.
ICALC=2, SECOND ORDER POLYNOMIAL. FIND MINIMUM OR
MAXIMUM.
ICALC=3, THIRD ORDER POLYNOMIAL. FIND MINIMUM AND MAXIMUM.
IF THEY ARE REAL.
A1, A2, A3 - COEFFICIENTS OF THE POLYNOMIAL,
Y = AZ + A1*X + A2*X**2 + A3*X**3.
NOTE - AZ IS NOT NEED IN CALCULATIONS, SO IS NOT BROUGHT

```

```

APP05800
APP05810
APP05820
APP05830
APP05840
APP05850
APP05860
APP05870
APP05880
APP05890
APP05900
APP05910
APP05920
APP05930
APP05940
APP05950
APP05960
APP05970
APP05980
APP05990
APP06000
APP06010
APP06020
APP06030
APP06040
APP06050
APP06060
APP06070
APP06080
APP06090
APP06100
APP06110
APP06120
APP06130
APP06140
APP06150
APP06160
APP06170
APP06180
APP06190
APP06200
APP06210
APP06220
APP06230
APP06240
APP06250
APP06260
APP06270

```

```

CCCCCCCCCCCCCCCCCCCC

```



```

C-----
C      INPUT - ARRAY CONTAINING N VALUES OF X, N.GE.1.
C      X(N) IS ASSUMED TO BE MONOTONICALLY INCREASING.
C      REQUIRED DIMENSION OF X IS N.
C      F(N) - ARRAY CONTAINING N FUNCTION VALUES CORRESPONDING TO
C      THE VALUES OF X.
C      REQUIRED DIMENSION OF F IS N.
C      XMULT, XADD - UPDATING CONSTANTS IF X MUST BE INCREASED TO
C      FIND AN UPPER BOUND.
C      UPDATE FORMULA IS XNEW = (1.+XMULT)*X(N) + XADD.
C      XMULT AND XADD MUST BE NON-NEGATIVE AND AT LEAST ONE OF THEM
C      MUST BE POSITIVE.
C      N - NUMBER OF X-F PAIRS ALREADY CONTAINED IN X AND F. N.GE.1.
C-----
C      OUTPUT - -1.0 IF F(N).GE.F(N-1) WHERE N.GE.2. THIS INDICATES
C      THAT X CONTAINS BOUNDS ON F.
C      XNEW = PROPOSED NEW X-VALUE IF F(N).LT.F(N-1) FOR N.GE.2 OR
C      IF N=1.
C      XNEW = (1.+XMULT)*X(N)+XADD.
C-----
C      SUBROUTINE ADS006 (X,F,N,XMULT,XADD,XNEW)
C      DIMENSION X(1),F(1)
C
C      XNEW=(1.+XMULT)*X(N)+XADD
C      IF (N.LE.1) RETURN
C      NM1=N-1
C      IF (F(N).LT.F(NM1)) RETURN
C      MINIMUM IS BOUNDED.
C      XNEW=-1.
C      RETURN
C      END
C-----
C      SUBROUTINE ADS007(XL,X1,X2,XU)
C      *****
C      ROUTINE TO CALCULATE INITIAL INTERIOR POINTS, X1 AND X2 FOR
C      FINDING THE MINIMUM OF A FUNCTION USING THE GOLDEN SECTION METHOD.
C      *****
C      BY G. N. VANDERPLAATS
C      NAVAL POSTGRADUATE SCHOOL, MONTEREY, CALIF.
C      AUG. 1980
C-----
C      INPUT
C      XL - LOWER BOUND ON SOLUTION.
C      XU - UPPER BOUND ON SOLUTION.
C-----
C      OUTPUT
C      X1 - FIRST INTERMEDIATE POINT. X1=GOLD*(XU-XL).

```


INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2. Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3. Department Chairman, Code 69 Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	2
4. Professor G. N. Vanderplaats, Code 69Vn Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	1
5. Professor D. Salinas, Code 69S1 Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	1
6. Lt. L. E. Madsen 457 Foam Street Monterey, California 93940	1